



UNICEUB – CENTRO UNIVERSITÁRIO DE BRASÍLIA
FAET – FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

Técnicas para Apresentação de Imagens Utilizando LEDs

ALESSANDRO DA SILVA FERREIRA

Brasília
2007

ALESSANDRO DA SILVA FERREIRA

Técnicas para Apresentação de Imagens Utilizando LEDs

Orientador: Prof. Aderlon Marcelino Queiroz

**Monografia apresentada ao Centro
Universitário de Brasília, para obtenção
do título de Bacharel em Engenharia da
Computação.**

**Brasília-DF
Julho de 2007.**

Monografia aprovada como requisito para a obtenção do título de Bacharel em Engenharia da Computação – Centro Universitário de Brasília. Comissão Examinadora formada pelos professores:

Prof. Aderlon Marcelino Queiroz, MSc
Orientador

Prof. Miguel A. B. G. Telles Jr, MSc
Banca

Prof. Fabiano Mariath D'Oliveira, MSc
Banca

Brasília, 02 de julho de 2007.

"A grandeza não consiste em receber honras, mas em merecê-las".

Aristóteles.

AOS MEUS PAIS E A MINHA ESPOSA.

AGRADECIMENTOS

Agradeço a todos que diretamente ou não, contribuíram para o desenvolvimento deste projeto, e em particular:

Ao Professor Aderlon Marcelino Queiroz, pela orientação, paciência, incentivo, cobrança e sugestões de conteúdos que enriqueceram este projeto.

Aos professores do Curso de Engenharia da Computação do Centro Universitário de Brasília – UniCEUB que são responsáveis por parte do sucesso profissional de todos os seus alunos.

Aos meus pais, Meireluce Fernandes da Silva e Rafael Jaime Ferreira, que sempre estiveram do meu lado, ajudando, incentivando, confortando e me fortalecendo para que eu conseguisse concluir com êxito este projeto.

A minha esposa Ana Lucia pela paciência, compreensão, carinho e apoio diário.

A minha tia Rosa que tanto me apoiou. Aos meus avós, que aqui, ou distante sempre olharam por mim.

Aos amigos Bruno Mesquita, Ivan Sampaio, Daniel B. Strehl, pelo apoio nesta reta final.

E, principalmente, a DEUS por ter permitido que mais uma porta se abrisse em minha vida.

RESUMO

O presente trabalho propõe o desenvolvimento de um painel para apresentação de imagens formadas digitalmente a partir da utilização de LEDs . Os LEDs utilizados contêm múltiplas cores para apresentar imagens ao observador. Foram analisadas técnicas de implementação de forma analógica e digital para controle do circuito e no decorrer do trabalho o protótipo do produto foi implementado com a melhor técnica, observando-se o paradigma de melhor custo e melhor benefício. Técnicas como o PWM e a utilização de amplificadores operacionais foram analisadas e implementadas. As imagens formadas para apresentação foram figuras geométricas simples, como triângulos e quadrados, onde as cores foram alternadas durante a mudança. Padrões de teste, como ligar todos os pontos de luz com cores alternadas e emissão da cor branca em todos os pontos foram apresentados.

Palavras chaves: Tela, Diodo Emissor de Luz (LED), Circuito Integrado Programável (PIC), Microcontroladores, Semicondutores.

ABSTRACT

The present paper proposes the development of a panel for presentation of images formed digitally from the use of LEDs. The utilized LEDs contain multiple colors to present images to the observer. Techniques of implementation of analogical and digital form for the controlling of the circuit were analyzed and during the work the prototype of the product was implemented with the best technique, observing the paradigm of better cost and benefit. Techniques as the PWM and the use of operational amplifiers were analyzed and implemented. The images formed for presentation were simple geometric figures, as triangles and squares, where the colors were alternated during the change. Models of test, as turn on all the points of light with alternating colors and emission of the white color in all the points were showed.

Key words: *Display, Light Emitting Diodes (LED), Programmable Integrated Circuits (PIC), Microcontroller, Semiconductor.*

SUMÁRIO

SUMÁRIO DE TABELAS.....	I
SUMÁRIO DE FIGURAS.....	II
LISTA DE EQUAÇÕES.....	IV
LISTA DE SÍMBOLOS.....	V
LISTA DE SIGLAS E ABREVIATURAS.....	VI
CAPÍTULO 1 – INTRODUÇÃO.....	1
CAPÍTULO 2 – REFERENCIAL TEÓRICO.....	3
2.1 – O olho humano.....	3
2.2 – História da comercialização dos LEDs.....	5
2.3 – Percepção humana da cor.....	7
2.4 – Sistemas de cor.....	7
2.5 – RGB.....	8
2.6 – Luz, comprimento de onda e cor.....	10
2.6.1 - Onda.....	11
2.6.2 - Luz.....	12
2.6.3 – Cor	13
2.7 – Cor e seus diferentes conceitos.....	14
2.8 - Características das cores.....	15
2.9 – A luz do LED.....	18
2.9.1 – Curvas de distribuição de intensidade luminosa.....	18
CAPÍTULO 3 – SEMICONDUTORES.....	22
3.1 - O Semicondutor	22
3.1.1 - Semicondutores intrínsecos.....	26
3.1.2 - Impurezas Doadoras.....	26
3.1.3 - Impurezas Receptoras.....	27
3.1.4 - Intervalo (Gap).....	28
3.1.5 - Camada de depleção.....	28
3.2 - O diodo emissor de luz (LED).....	30
CAPÍTULO 4 – Modulação por Largura de Pulso.....	34
4.1 - O que é PWM.....	34
4.1.1 - Filtro DC.....	35
4.2 – Comparando PWM com variação de tensão em LEDs.....	36
CAPÍTULO 5 – PIC.....	38
5.1 - OSCILADORES.....	38
5.1.1 - Oscilador do tipo RC (Resistor-Capacitor).....	38
5.1.2 - Oscilador do tipo Ressonador.....	39
5.1.3 - Oscilador do tipo Cristal.....	41
5.1.4 - Oscilador Interno ao PIC.....	42
5.1.5 - Outros Osciladores (Circuitos de Oscilação).....	43
CAPÍTULO 6 – MONTAGEM.....	44
6.1 - O Circuito Analógico	44
6.1.1 - Primeira etapa.....	45
6.1.1.1 - Conclusões da primeira etapa.....	47
6.1.2 - Segunda etapa (procedimento e conclusões).....	47
6.1.3 - Conclusões gerais do circuito analógico.....	48

6.2 - O Circuito Digital.....	49
6.2.1 - O resistor de acoplamento do LED.....	50
6.2.2 - A paleta de cores CGA.....	50
6.2.3 - A implementação do circuito digital	52
6.2.3.1 - Primeira etapa.....	53
6.2.3.1.1 - Conclusões da primeira etapa.....	54
6.2.3.2 - Segunda etapa.....	55
6.2.3.2.1 - Conclusões da segunda etapa.....	56
6.2.3.3 - Terceira etapa.....	56
6.2.3.3.1 - Conclusões da terceira etapa.....	57
6.3 – Comparações entre as análises.....	57
6.4 - O Protótipo da Tela de LEDs.....	58
6.4.1 - Primeira Etapa.....	59
6.4.2 - Segunda Etapa.....	60
6.4.3 - Terceira Etapa.....	62
6.4.4 - Quarta Etapa.....	63
6.4.5 - Conclusões da montagem do Protótipo.....	64
<i>CAPÍTULO 7 – CONCLUSÃO.....</i>	<i>66</i>
7.1 – Conclusões.....	66
7.2 – Dificuldades encontradas.....	66
7.3 – Trabalhos futuros.....	67
<i>Referências Bibliográficas.....</i>	<i>68</i>
<i>Apêndice 1.....</i>	<i>70</i>
<i>Apêndice 2.....</i>	<i>72</i>
<i>Apêndice 3.....</i>	<i>74</i>
<i>Apêndice 4.....</i>	<i>79</i>
<i>Apêndice 5.....</i>	<i>86</i>
<i>Anexo 1.....</i>	<i>88</i>

SUMÁRIO DE TABELAS

<i>Tabela 1 - Sistemas de Cores.....</i>	<i>8</i>
<i>Tabela 2 – Número de elétrons por órbita.....</i>	<i>24</i>
<i>Tabela 3 – Energia necessária para salto.....</i>	<i>28</i>
<i>Tabela 4 – Níveis de Brilho em Hexadecimal.....</i>	<i>51</i>
<i>Tabela 5 – Resumo das conclusões de análises dos circuitos.....</i>	<i>57</i>
<i>Tabela 6 – Correspondência de pinos com LEDs.....</i>	<i>62</i>

SUMÁRIO DE FIGURAS

<i>Figura 1 – Olho humano em corte.....</i>	<i>3</i>
<i>Figura 2 – Estrutura da retina humana.....</i>	<i>4</i>
<i>Figura 3 – Cores Aditivas (RGB).....</i>	<i>9</i>
<i>Figura 4 – Tríade em CRT.....</i>	<i>10</i>
<i>Figura 5 – Espectro eletromagnético.....</i>	<i>12</i>
<i>Figura 6 – Prisma.....</i>	<i>14</i>
<i>Figura 7 – Matiz, Brilho e Saturação.....</i>	<i>15</i>
<i>Figura 8 – Duplo cone de Ostwald.....</i>	<i>16</i>
<i>Figura 9 – Ordenação de Munsell.....</i>	<i>16</i>
<i>Figura 10 – Comprimento de onda da cor branca e colorida.....</i>	<i>17</i>
<i>Figura 11 – Mistura de cores.....</i>	<i>17</i>
<i>Figura 12 – Triângulo CIE.....</i>	<i>18</i>
<i>Figura 13 – Representação da curva de distribuição luminosa de uma lâmpada incandescente.....</i>	<i>19</i>
<i>Figura 14 – Curva Fotométrica LED 30°.....</i>	<i>19</i>
<i>Figura 15 – Curva Fotométrica LED 15°.....</i>	<i>20</i>
<i>Figura 16 – LED emitindo e Detalhe.....</i>	<i>21</i>
<i>Figura 17 – Órbitas eletrônicas.....</i>	<i>23</i>
<i>Figura 18 – Polarização Reversa.....</i>	<i>29</i>
<i>Figura 19 – Diodos axiais.....</i>	<i>29</i>
<i>Figura 20 – Estrutura típica do chip de um diodo emissor de Luz (LED).....</i>	<i>31</i>
<i>Figura 21 – Formação de cones de luz no chip de LED.....</i>	<i>32</i>
<i>Figura 22 – Representação de um pulso em PWM.....</i>	<i>35</i>
<i>Figura 23 – Filtro para PWM atingir nível DC.....</i>	<i>36</i>
<i>Figura 24 – Variando Tensão do LED Nichia.....</i>	<i>37</i>
<i>Figura 25 – Oscilador do tipo RC.....</i>	<i>39</i>
<i>Figura 26 – Ressonador de 2 Pinos.....</i>	<i>40</i>
<i>Figura 27 – Ressonador de 3 Pinos.....</i>	<i>40</i>
<i>Figura 28 – Cristal de Quartzo.....</i>	<i>42</i>
<i>Figura 29 – Circuito para variação de brilho do LED.....</i>	<i>45</i>
<i>Figura 30 – Implementação do Circuito com AMPOP.....</i>	<i>46</i>
<i>Figura 31 – Análise saída AMPOP pelo osciloscópio.....</i>	<i>47</i>
<i>Figura 32 – Análise 3 LEDs analógico.....</i>	<i>48</i>
<i>Figura 33 – Relação de 16 cores em Hexadecimal.....</i>	<i>52</i>
<i>Figura 34 – Variação da intensidade de brilho com PWM.....</i>	<i>54</i>
<i>Figura 35 – Circuito para a terceira etapa de análise do circuito digital.....</i>	<i>56</i>
<i>Figura 36 – Esquemático do display 3x3 colorido.....</i>	<i>60</i>

<i>Figura 37 – Protótipo Final.....</i>	<i>61</i>
<i>Figura 38 – Protótipo Final Ligado.....</i>	<i>63</i>
<i>Figura 39 – Triade emitindo cor marrom.....</i>	<i>64</i>

LISTA DE EQUAÇÕES

Equação 1..... 10

Equação 2..... 11

Equação 3..... 23

Equação 4..... 34

Equação 5..... 34

Equação 6..... 35

Equação 7..... 35

Equação 8..... 36

Equação 9..... 36

Equação 10..... 50

Equação 11..... 50

LISTA DE SÍMBOLOS

f : Frequência de onda eletromagnética

c : Velocidade da luz no vácuo

λ : (lambda) – Comprimento de onda

E : Amplitude de onda

i : Corrente elétrica em ampères

A : Ampères

$1e$: um elétron

eV : Elétron Volt

C : Coulomb

K : Catodo

E_c : Energia do elétron na banda de condução

E_g : Energia do elétron necessária para vender o intervalo (*gap*).

s : Segundo, unidade de medida de tempo.

lum : Lumens, unidade de medida do fluxo luminoso.

g : grama

F : Farad

\AA : ångström, onde $1 \text{\AA} = 10^{-10} \text{ m}$

nm : nanômetro, onde $1nm = 10^{-9}m$

μ : micro, onde $1\mu = 10^{-6}$

Ω : ohm, unidade de medida de resistência elétrica.

LISTA DE SIGLAS E ABREVIATURAS

AMPOP: Abreviação de Amplificador Operacional

CMYK: Acrônimo significando – **C**iano (do dicionário Aurélio, azul esverdeado), **M**agenta (do dicionário Aurélio, carmim ou rosa choque), **Yellow** (amarelo) e **K** (*black*) (preto) representando o preto. Cores primárias para impressão.

CRT: Acrônimo do inglês – *Cathode Ray Tube*, Tubo de Raios Catódicos.

DDP: Diferença de Potencial Elétrico

DLP: Acrônimo do inglês - Digital Light Processing (processamento digital de luz), tecnologia desenvolvida pela empresa Texas Instruments, para projetores de imagem em 1987.

IEEE: Instituto de Engenheiros Eletricistas e Eletrônicos (*Institute of Electrical and Electronics Engineers*).

LCD: Acrônimo do inglês – *Liquid Crystal Display*, Tela de Cristal Líquido.

LED: *Light Emitting Diodes*, Diodo emissor de Luz

NTSC: Acrônimo do inglês – *National Television Standards Committee*, Comitê Nacional de Padrões Televisivos.

Pixel: Elemento de Imagem (*Picture Element*).

Protoboard: Placa com contatos metálicos destinada a construção de protótipos eletrônicos.

PWM: *Pulse Width Modulation* - modulação por largura de pulso

RGB: Sistema de cores vem do Inglês *Red, Green and Blue* (Vermelho, Verde e Azul).

Tríade: Representação de um pixel pelas cores RGB

YIQ: Espaço de cor onde Y representa a luminância e, I e Q representam a cromaticidade.

GaAsP: Composto químico da união dos elementos Gálio, Arsênio e Fósforo.

AlGaInP: Composto químico da união dos elementos Alumínio, Gálio, Índio e Fósforo.

GaN: Composto químico da união dos elementos Gálio e Nitrogênio.

6H-SiC: Composto químico da união dos elementos Hidrogênio, Silício e Carbono.

AlN: Composto químico da união dos elementos Alumínio e Nitrogênio.

GaAs: Composto químico da união dos elementos Gálio e Arsênio.

AlInP: Composto químico da união dos elementos Alumínio, Índio e Fósforo.

GaP: Composto químico da união dos elementos Gálio e Fósforo.

CAPÍTULO 1 – INTRODUÇÃO

Muitas foram as motivações para criar este projeto e a falta de conhecimento de como alguma coisa opera é para muitos a mola mestra para o desenvolvimento de um produto. Em alguns casos, este produto acaba se tornando melhor que o antecessor.

Criar uma tela (*display*) de LEDs que possibilitasse mostrar diversas cores com a utilização de LEDs e que, se possível, apresenta-se a possibilidade de acoplar diversas destas mesmas telas umas às outras com o intuito de formar uma tela maior. Este é um objetivo difícil de ser alcançado, principalmente imaginando que o custo de uma tela de 320x240 pontos coloridos para apresentação de imagens em grandes ambientes externos pode chegar à soma de US\$ 80,000.00 (oitenta mil dólares), conforme consulta verbal ao Sr. André Brandão proprietário de empresa de mídias publicitárias.

Mas em nível acadêmico e com o pouco tempo e recurso disponível para criação deste produto implementaremos uma tela de 3 pontos (*pixels*) de altura por 3 pontos (*pixels*) de largura garantindo modularidade e a melhor relação custo e benefício desta implementação. Outra meta importante é que, se pretendemos criar um produto com a finalidade de apresentar imagens, devemos imaginar que ele possa ser utilizado em locais pequenos também. Desta forma, a utilização de válvulas está descartada. Para aumentar a escalabilidade será necessário diminuir ao máximo o tamanho do circuito, sua emissão de calor e, se possível, diminuir seu consumo elétrico.

Nos capítulos que se seguem, inicialmente faremos uma explanação de temas importantes para o desenvolvimento deste trabalho. No segundo capítulo, apresentaremos alguns conceitos sobre como enxergamos, quais foram os primeiros estudos sobre a luz, suas características, como ela se divide em cores, os primeiros equipamentos que utilizavam o conceito de tríade de cores, e outras características.

No terceiro capítulo, apresentaremos como é formada a luz que emana dos LEDs. Para isto discutiremos seus princípios eletrônicos e ópticos.

A partir do quarto capítulo, entraremos na parte prática do projeto, analisando qual deve ser a abordagem de construção, os componentes a utilizar, as técnicas eletrônicas que serão implementadas. E por fim, no sexto capítulo, apresentaremos a montagem do protótipo e de seus precursores.

Finalmente, no sétimo capítulo apresentaremos as conclusões deste trabalho e as sugestões para a continuação do mesmo.

CAPÍTULO 2 – REFERENCIAL TEÓRICO

É necessário explicar primeiramente como o ser humano consegue entender as informações visuais e as interpretar.

Fazendo uma comparação com o mundo digital dos cinco sentidos, a visão seria a que utiliza a maior largura de banda do cérebro. A quantidade de informação percebida pelos olhos é imensa, sendo em grande parte descartada. Somente uma pequena parte desta infinidade de informação é realmente armazenada e processada [KOCH, 2007].

Um dos pilares deste projeto se baseia nesta característica, que será melhor explicada no capítulo a seguir.

2.1 – O olho humano

A função de capturar a imagem e converter as informações em impulsos elétricos no corpo humano é do globo ocular, como representado em corte na figura 1. As imagens incidentes na porção frontal do olho, são filtradas com relação ao fluxo luminoso e foco, e são projetadas na porção interna traseira do olho denominada retina.

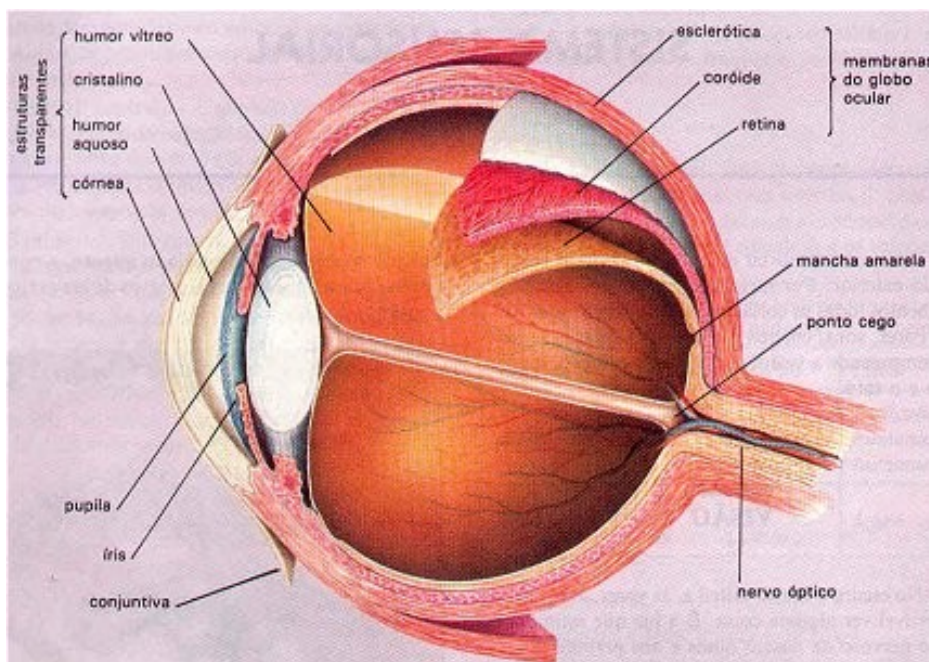


Figura 1 – Olho humano em corte.

[Fonte: <http://www.corpohumano.hpg.ig.com.br/variados/thaty/tathy.html>, modificado]

Conforme podemos observar na figura 2 que representa a estrutura axial da retina (vista em detalhe do olho na figura 1), na retina, os bastonetes e os cones (figura 2 - à direita, 1 cone e 9 bastonetes; ao centro, 2 células bipolares; à esquerda, 3 axônios de células ganglionares que pertencem ao nervo óptico) são responsáveis pela diferenciação de luzes escuras e claras, respectivamente. Os bastonetes existem em menor número e os cones existem em maior número. Conforme explicaremos no subitem 2.8 deste capítulo, cores, comumente, são relacionadas a luzes com pouco brilho, pois atingem poucas camadas do espectro visível, enquanto o branco, mesmo que pouco intenso, sempre está relacionado à luz clara, por conter todas as outras cores.

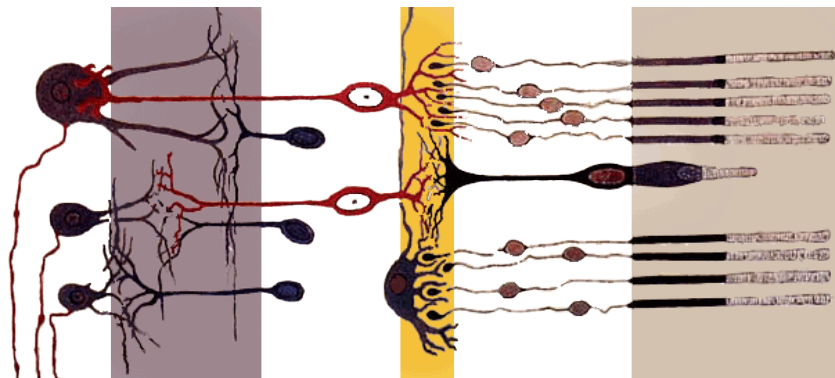


Figura 2 – Estrutura da retina humana
[Fonte: Cajal, 1911, modificado]

De forma simplificada, podemos classificar os cones em 3 tipos, os do tipo B (do inglês Blue) que captam frequências que abrangem as cores azul e violeta, os do tipo G (do inglês Green) que captam frequências que abrangem as cores verde e amarela e os do tipo R (do inglês Red) que são sensibilizados pelas cores vermelha e laranja (figura 2). Os três tipos de cones são distribuídos de forma não uniforme na área central da retina, de forma que somente 6% dos mesmos são do tipo B, por questões evolutivas.

O físico belga Joseph Antoine Ferdinand Plateau foi quem mediu pela primeira vez o tempo da persistência retiniana, em meados de 1830. Esta teoria consiste na capacidade da retina em manter, por uma fração de segundo, uma imagem, mesmo após esta imagem haver mudado.

Mesmo depois do cérebro ter recebido os impulsos da retina, a mesma continua mandando informações, por aproximadamente 1/10 de segundo, após o último estímulo luminoso. Esta disfunção permite que tenhamos a sensação de movimento entre a sobreposição de imagens. O cinema utiliza esta característica até os dias atuais. [SALLES, 2007]

2.2 – História da comercialização dos LEDs

A procura de materiais e estruturas para a fabricação de LEDs (Light Emitting Diodes, Diodo Emissor de Luz) mais eficientes levaram este dispositivo a emitir comprimentos de onda além do que o olho humano pode identificar. Atualmente, devido a aplicações de engenharia em escala atômica, a eficiência dos LEDs de alto brilho pode ser comparada às eficiências das lâmpadas de halogênio e as de tungstênio incandescentes [STRINGFELLOW, 1997].

A pesquisa de materiais emissores na região do espectro visível começou em meados de 1960 com a utilização dos semicondutores Germânio (Ge) e Silício (Si) que não apresentavam emissões visíveis. Os mesmos apresentavam emissões infravermelhas. Com a utilização da mistura de elementos que geravam ligas com propriedades de semiconductividade foi possível a obtenção de LEDs que emitiam fótons suficientes para a obtenção de algumas cores como o vermelho, o verde e o amarelo. Esta emissão não era suficiente para, por assim dizer fazer uma iluminação, somente servia para criar LEDs indicativos.

Duas abordagens feitas por diferentes laboratórios para a obtenção de LEDs de alto brilho foram as precursoras destes tipos de semicondutores. A primeira foi a adição de impurezas isoeletrônicas, que são “...substâncias que não contribuem para a condutividade

elétrica de um semicondutor, mas podem influenciar nas propriedades luminescentes do material, atuando como centros de recombinação de elétrons excitados e buracos.” [GURGEL, 2005], com o objetivo de direcionar a emissão de fótons. Isto melhorou a eficiência em 15% para os LEDs vermelhos. Foi utilizado um isótopo Zn-O (Zinco-Oxigênio) como dopante do composto semicondutor GaP (Gálio – Fósforo) [STRINGFELLOW, 1997], que gerava emissões com comprimento de onda de aproximadamente 700nm, que causava uma sensibilidade visual ainda baixa. A segunda abordagem envolve a utilização de uma liga composta de materiais para aumentar a vazão energética da liga binária dentro da região do espectro visível. Um exemplo é a adição do Fósforo no GaAs (Gálio – Arsênio) que produz uma emissão na faixa do vermelho (GaAsP) [STRINGFELLOW, 1997] de eficiência aproximada de 0,1%. Nesta mesma técnica, nos dias atuais, chegamos a utilizar ligas de 4 (quatro) elementos, como por exemplo AlGaInP (Alumínio Gálio Índio Fósforo) que produzem emissões completas na faixa do vermelho ao verde e com uma eficiência muito maior.

Ainda assim a obtenção do azul era um problema. No final da década de 70 a cor azul já era utilizada dentro dos tubos de raios catódicos, porém, quando os mesmos elementos eram transpostos para junções p-n os mesmos tendiam a se estabilizar. Foi com a utilização de novas técnicas de produção que se conseguiu uma melhor pureza das junções, sendo assim possível a obtenção da cor azul [STRINGFELLOW, 1997] porém, a mesma ainda não poderia ser comercializada, pois sua molécula era muito grande e de difícil produção (6H-SiC) o que encarecia o produto. Os únicos semicondutores capazes de produzir realmente a cor azul eram os nitrosos.

Na maioria das tentativas de se criar ligas de nitrogênio com o fósforo ou o arsênio havia uma tendência de auto estabilização do material, o que inviabilizava sua produção. Foi em 1989 que se tentou a união de GaN para a obtenção de luminescência na faixa do azul

[STRINGFELLOW, 1997]. As pesquisas mais atuais estão misturando o GaN com o AlN para melhorar o desempenho dos LEDs azuis.

2.3 – Percepção humana da cor

O conceito de cor não pode ser explicado pois a cor é uma propriedade e uma percepção inerente a quem vê. Ela é captada pela retina e interpretada pelo cérebro e tem valor de emoção.

“ Pelo fato da cor ser uma percepção e não um conceito científico, não existe uma teoria que responda de modo satisfatório o conceito do que vem a ser cor, entretanto, para podermos estudar cor, podemos considerá-la um fenômeno de luz e de visão. Ela não existe por si só. No mundo físico, a cor é uma das propriedades que caracterizam a aparência das coisas que vemos.”
[SILVESTRE, 2005]

2.4 – Sistemas de cor

Existem basicamente 2 (dois) sistemas de cores mais usados, o RGB¹ e o CMYK². São chamados de sistema pois há um modelo matemático para se chegar aos mesmos. Para saber qual dos dois estamos utilizando precisamos conhecer o método de criação que formou a cor referenciada. Cabe acrescentar que os sistemas RGB e CMYK não são os únicos desenvolvidos, existem outros como o YIQ³ que é a base de desenvolvimento do NTSC⁴ e outros.

¹ Acrônimo do inglês – *Red, Green and Blue*, cores primárias para luz.

² Acrônimo do inglês – *Ciano, Magenta, Yellow* e *K* representando o preto. Cores primárias para impressão.

³ Espaço de cor onde Y representa a luminância e, I e Q representam a cromaticidade.

⁴ Acrônimo do inglês – *National Television Standards Committee*

A união de cores forma outras cores. Àquelas cores com as quais é possível formar todas as outras são chamadas de cores primárias e a mistura de duas cores primárias é denominada cor secundária, sendo que as cores primárias variam entre sistemas de cor.

Utilizando a “cor branca” como referência podemos dividir o conjunto de cores como cores aditivas ou cores subtrativas. As cores aditivas são as que não dependem da incidência de uma luz branca para se formar, a união das mesmas é que forma a cor branca. As cores que dependem da incidência de uma luz branca para serem identificadas são chamadas de subtrativas. Na ausência de luz uma cor subtrativa não pode ser vista. As cores aditivas, quando unidas, formam a luz de cor branca (sendo este último um dos pilares deste trabalho). Podemos verificar, na tabela 1 a exemplificação comparativa dos dois sistemas.

Tabela 1 - Sistemas de Cores						
	PRIMÁRIAS			SECUNDÁRIAS		
CORES ADITIVAS (Luz)	VERMELHO	VERDE	AZUL	CIANO	MAGENTA	AMARELO
				=	=	=
				VERDE	VERMELHO	VERMELHO
				+	+	+
CORES SUBTRATIVAS (Pigmentos)	CIANO	MAGENTA	AMARELO	AZUL	AZUL	VERDE
				VERMELHO	AZUL	VERDE
				=	=	=
				MAGENTA	CIANO	CIANO
				+	+	+
				AMARELO	MAGENTA	AMARELO

[Fonte: [SILVESTRE, 2005]]

2.5 – RGB

O sistema de cores aditivas RGB é formado pela união das cores vermelha, do inglês *Red*, da cor verde, do inglês *Green* e da cor azul, do inglês *Blue*. Seu nome é formado pelo acrônimo das letras iniciais das cores em inglês.

A figura 3 nos mostra o sistema RGB imaginando 3 fontes de luz independentes.

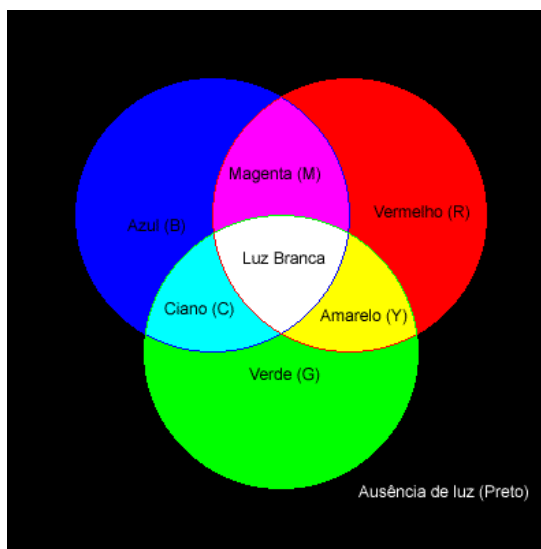


Figura 3 – Cores Aditivas (RGB)

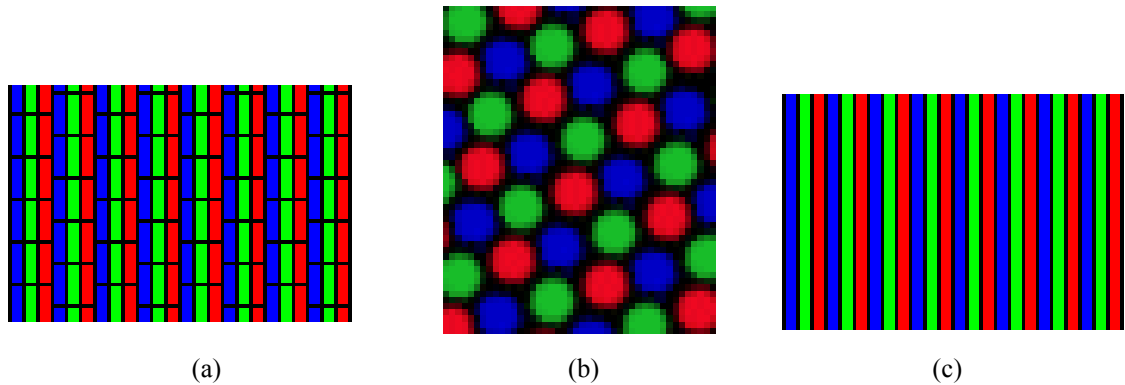
[Fonte: [SILVESTRE, 2005]]

Exemplificando, este sistema de cores pode ser encontrado em dispositivos que emitem luz, como a televisão, o monitor do computador, tanto os novos com tecnologia LCD⁵ como os antigos CRT⁶, e também em projetores que utilizam a tecnologia LCD. Os projetores que utilizam tecnologia DLP⁷ formam o branco de outra forma que não é foco deste trabalho. Os monitores CRT produzem cores ao enviarem feixes elétricos para pontos de pequenas substâncias fosforescentes vermelhas, verdes e azuis, que ficam atrás do vidro do monitor. As substâncias RGB são agrupadas em tríades, conforme visto na figura 4, ou organizadas em retângulos unidos lateralmente. A variação da quantidade de elétrons que atingem os compostos fosforescentes (vermelho, verde e azul) na porção interna do tubo fazem com que haja variação na intensidade de cada cor.

⁵ Acrônimo do inglês – *Liquid Crystal Display*

⁶ Acrônimo do inglês – *Cathode Ray Tube*

⁷ Acrônimo do inglês - *Digital Light Processing*, tecnologia desenvolvida pela empresa *Texas Instruments*, para projetores de imagem em 1987.



(a) Tubos de imagem tipo Cromaclear da fabricante Nec, (b) Tubo de imagem FST-Invar (Flat Square Tube) e (c) Tubo de imagem Trinitron do fabricante Sony.

Figura 4 – Tríade em CRT

[Fonte: <http://www.commentcamarche.net/pc/ecran-crt.php3>]

Estes tipos de dispositivos usam apenas três cores para criar todas as outras cores. Em uma imagem colorida, a representação de cores pode ser encontrada por intermédio da equação 1, onde r, g, b são coeficientes da mistura correspondente às intensidades de cada canal R, G e B [SILVESTRE, 2005].

$$C = r.R + g.G + b.B \quad (\text{Equação 1})$$

“As cores secundárias neste sistema são sempre mais claras que as primárias, pois quando se adiciona uma luz à outra, o resultado é uma luz mais clara. Com isso podemos mostrar o efeito inverso do prisma ... “[SILVESTRE, 2005], conforme a autora afirma, a experiência de Newton com o prisma que separa a cor branca em vários feixes coloridos pode ser aplicada de forma inversa se utilizarmos luzes com as cores primárias direcionadas ao mesmo ponto de reflexão. Temos então, como resultado de observação, a luz branca.

2.6 – Luz, comprimento de onda e cor

Determinados aspectos dos conceitos do que é luz, comprimento de onda e cor são muito complexos e não são extremamente fundamentais para o escopo deste trabalho,

portanto apresentaremos de forma sucinta estes conceitos e como os mesmos se relacionam. A óptica é a área da ciência que mais estuda a luz, já que a mesma apresenta comportamento de onda e de partícula.

2.6.1 - Onda

Como explica o estudo de ondas em física, as ondas transportam energia e momento [YOUNG, 2003], sem transportar matéria. Em nosso estudo utilizaremos as ondas eletromagnéticas, pois são as que melhor descrevem o comportamento da luz como onda.

Nos estudos de física aprendemos que todas as ondas eletromagnéticas se propagam no vácuo com velocidade constante $c \approx 3 \times 10^8$ m/s. Conhecendo c e o valor da frequência f de uma onda eletromagnética podemos determinar o comprimento de onda λ desta radiação. A equação 2 demonstra a fórmula da onda [BERTULANI, 2007].

$$\lambda = c/f \qquad \text{(Equação 2)}$$

Podemos citar alguns exemplos de ondas eletromagnéticas e como as mesmas são utilizadas na atualidade, são elas: a luz visível (faixa de 400 até 700 nm), as ondas de rádio (faixa de 20 cm até 10^5 m), os raios-x (faixa de 10^{-1} até 10 Å), as ondas ultravioletas (faixa de 1 até 400 nm) e o infravermelho (faixa de 700 μ m até 1 mm) [SILVESTRE, 2005]. Estes diferentes tipos de ondas eletromagnéticas só se diferenciam pelo seu comprimento de onda e frequência como observadas na figura 5.

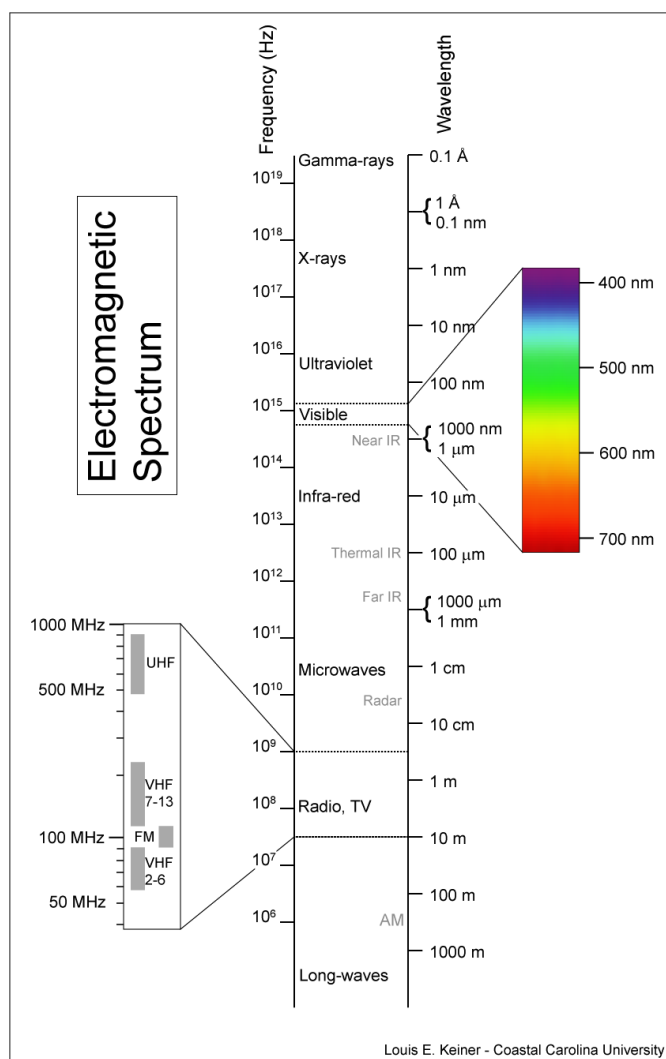


Figura 5 – Espectro eletromagnético

As cores primárias da luz são formadas por um único comprimento de onda (levar em consideração brilho, matiz e saturação), enquanto as demais cores são produzidas pela mistura de vários comprimentos de onda e a luz branca pode conter todos os comprimentos de onda, dependendo da percepção humana.

2.6.2 - Luz

O campo da física que estuda o comportamento da luz e suas aplicações, como sua propagação, natureza e fenômenos é a óptica.

Somente podemos perceber o mundo que nos cerca graças à luz. Observamos que a cor é um fenômeno que ocorre com a luz, e sua forma de percepção varia entre os seres humanos.

Estudos mostram que a luz, por vezes, se comporta como partícula e em outras ocasiões se comporta como onda. Estas características são estudadas desde o início dos tempos e até os dias atuais não se sabe por completo como são formados todos os efeitos luminosos. Esta teoria que apresenta os dois comportamentos estudados na luz é conhecida como dualidade onda-partícula.

“A dualidade onda-partícula diz que a luz é formada por pequenas partículas, fótons, que se propagam em grande velocidade, porém em linha reta. Diz também que a luz parte de uma fonte luminosa em movimento ondulatório, e por ser uma onda eletromagnética, não precisa de nenhum meio para se propagar”[SILVESTRE, 2005].

2.6.3 – Cor

Foi Isaac Newton o primeiro a demonstrar que a luz branca que passa através de um prisma de vidro não recebe partículas de outras cores e sim obriga as diferentes cores da luz branca a se propagarem em ângulos diferentes [FREEDMAN, 2003]. Desta forma, podemos concluir que a luz branca é formada pela composição de um grupo de luzes coloridas. Como visto anteriormente, a luz branca, no espectro eletromagnético, é formada por um grupo de ondas que podemos chamar de luz visível.

Na representação da figura 6 podemos identificar as cores que são formadas com a dispersão da luz branca em diferentes comprimentos de onda.

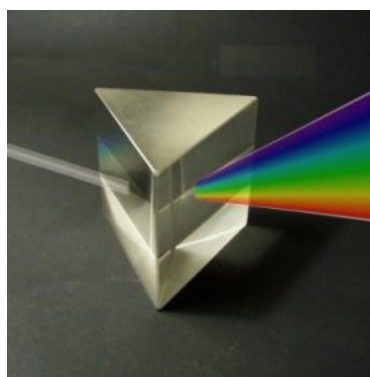


Figura 6 – Prisma
[Fonte: SILVESTRE, 2005]

2.7 – Cor e seus diferentes conceitos

Podemos definir cor como “... a sensação de um observador cuja retina é estimulada por energia radiante.” [SILVESTRE, 2005] esta definição parece pobre mas ela realmente é correta, pois, apesar de podermos criar padrões para as cores elas são vistas e interpretadas pelo observador. Um exemplo claro desta situação é o defeito genético encontrado em pessoas daltônicas⁸ que não conseguem distinguir entre as cores verde e vermelha, interpretando-as de forma trocada, em relação à maioria dos seres humanos.

A percepção das cores ocorre, quando luzes com comprimentos de onda diferentes entram em nossos olhos e estimulam milhares de células de nossa retina. No subitem 2.1 deste capítulo foram mencionados os cones e bastonetes que se encontram na retina do olho humano. Cones e bastonetes são células sensíveis à luz. Enquanto os bastonetes só são sensibilizados entre luzes claras e escuras, os cones são sensíveis a todos os comprimentos de ondas de luz formados pelas cores primárias Vermelho, Verde e Azul (RGB).

⁸ Daltonismo - Incapacidade de diferenciar certas cores, em especial o vermelho, donde a impossibilidade de distinguir, p. ex., o vermelho do verde. [Cf., nessas acepç., discromatopsia.] [Aurelio XXI].

2.8 - Características das cores

Para representar uma cor, os teóricos que estudam esta característica da luz tão subjetiva criaram alguns conceitos para melhorar sua determinação.

Para misturar o azul ao verde e ao vermelho e se obter a cor branca é desnecessário dizer que o brilho de cada cor deve ser regulado. Se isto não for feito podemos ter um vermelho mais intenso que o azul e o verde e obtemos uma cor resultante mais violeta. Desta forma existem características como o brilho que é o aumento da amplitude do comprimento de onda da cor.

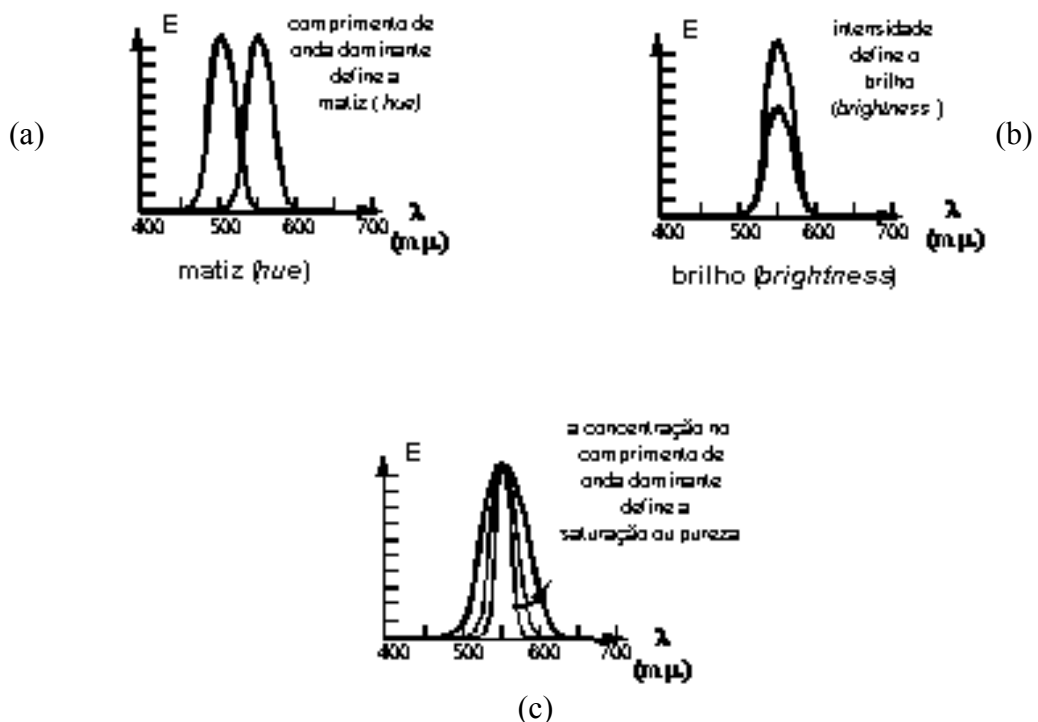


Figura 7 – Matiz, Brilho e Saturação

[Fonte: <http://www.if.ufrj.br/teaching/luz/cor.html>]

Quando existe a mistura de um comprimento de onda com outro, mesmo que seja próximo, surgirá uma nova cor que apresentará um comprimento de onda dominante em relação as duas primeiras. A esta característica damos o nome de matiz. A concentração da cor dominante chamamos de saturação ou pureza da cor (menos saturadas aproximam-se do

cinza. Na figura 7(c) , quanto mais fina a curva mais pura). Para representar essa classificação levando em conta as cores primárias, secundárias, misturas, saturação, brilho e matiz, elaboraram-se vários esquemas, dentre eles destacam-se o duplo cone de Ostwald, Figura 8, e a ordenação de Munsell, Figura 9. Essas representações nos exemplificam como funcionam essas medidas.



Figura 8 – Duplo cone de Ostwald
[Fonte: SILVESTRE, 2005]

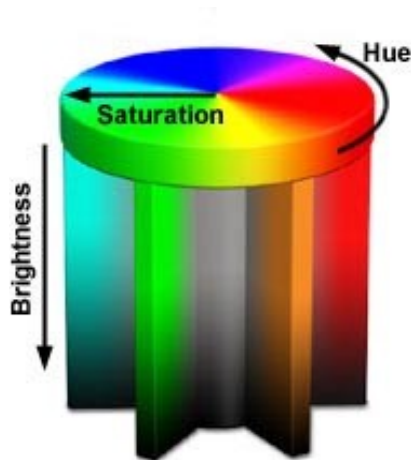


Figura 9 – Ordenação de Munsell
[Fonte: SILVESTRE, 2005]

Na figura 10, podemos entender a diferença, no gráfico dado, entre o comprimento de onda (λ) e a amplitude da onda (E) que o branco contém as outras três cores primárias adicionadas de pouca saturação de cada cor, muita matiz de todas as cores e o brilho depende do observador.

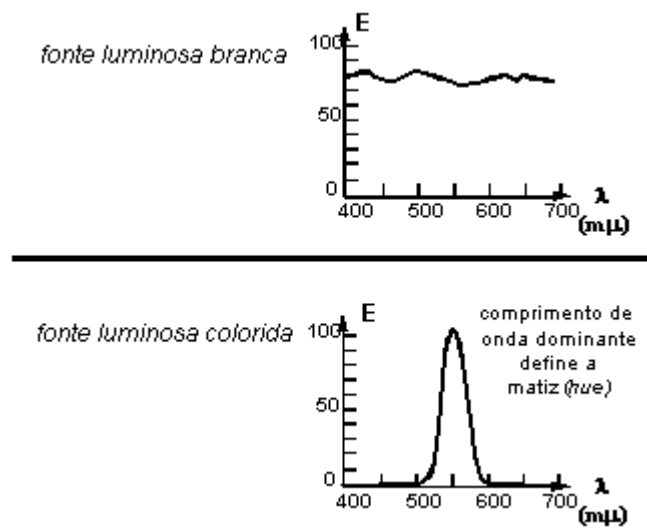


Figura 10 – Comprimento de onda da cor branca e colorida.
 [Fonte: <http://www.if.ufrj.br/teaching/luz/cor.html>]

A título de exemplificação, se misturarmos duas fontes de cores o resultado no gráfico será aproximado ao gráfico dado na figura 11 (E_{a+b}).

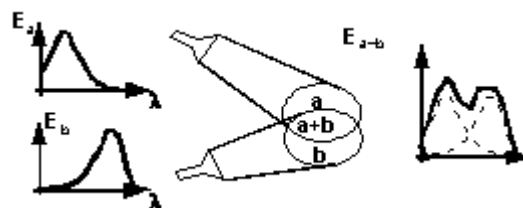


Figura 11 – Mistura de cores.
 [Fonte: <http://www.if.ufrj.br/teaching/luz/cor.html>]

Mas o tipo de representação mais utilizado para fazer a interpretação das cores, suas intensidades e outros cálculos é o Triângulo CIE (figura 12), apresentado abaixo. Sua base contém as cores relacionadas com o vermelho.



Figura 12 – Triângulo CIE.

[Fonte: <http://repositorium.sdum.uminho.pt/bitstream/1822/940/10/Texto-cap%C3%ADtulo+3.pdf>]

2.9 – A luz do LED

Utilizando-se o manual de especificações (*datasheet em anexo*) de um LED fabricado pela empresa japonesa Nichia Corporation podemos analisar as características de iluminação de um LED específico e extrapolar para outros LEDs, utilizando os conhecimentos das curvas de intensidade luminosa.

2.9.1 – Curvas de distribuição de intensidade luminosa

Podemos representar graficamente as várias medições de intensidades luminosas de uma fonte luminosa imaginando sua fonte no centro de uma esfera imaginária. A assim denominada curva fotométrica, normalmente, é fornecida pelos fabricantes de lâmpadas em seus catálogos de produtos para cada produto. Trata-se de um diagrama polar no qual se considera a lâmpada ou a luminária reduzida a um ponto no centro do diagrama, em que se representa a intensidade luminosa nas várias direções por vetores, partindo do centro do diagrama. A curva de distribuição da intensidade luminosa é obtida ligando-se as extremidades destes vetores. [CUSINATO, 2006]

Por convenção, os diagramas são elaborados para 1000 lumens⁹, com o objetivo de permitir a comparação de lâmpadas diferentes e observando-se que existem valores diferentes para o fluxo inicial de cada lâmpada.

A distribuição de luz de uma lâmpada incandescente é praticamente uniforme, imaginando que a mesma esteja ligada sem um bocal e sem luminária. Na figura 13, podemos observar um gráfico da superfície fotométrica de uma lâmpada incandescente.

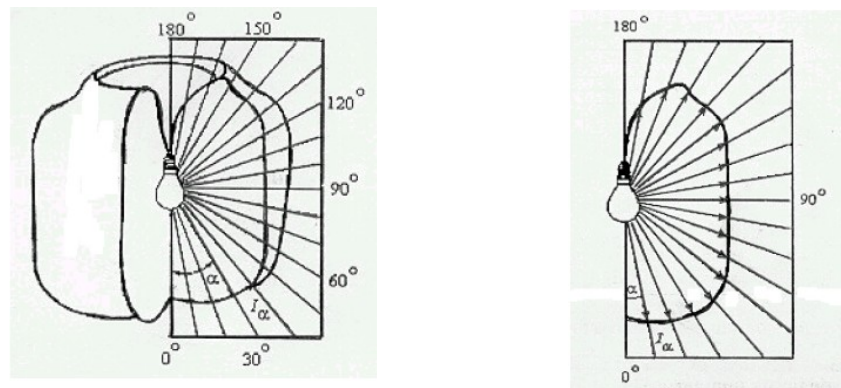


Figura 13 – Representação da curva de distribuição luminosa de uma lâmpada incandescente

[Fonte: [CUSINATO, 2006]]

Observamos porém que lâmpadas fluorescentes tubulares possuem duas curvas que representam sua intensidade luminosa que são causadas por seu formato assimétrico. Agora, podemos comparar a curva de distribuição luminosa de uma lâmpada incandescente com as curvas fotométricas de um LED.

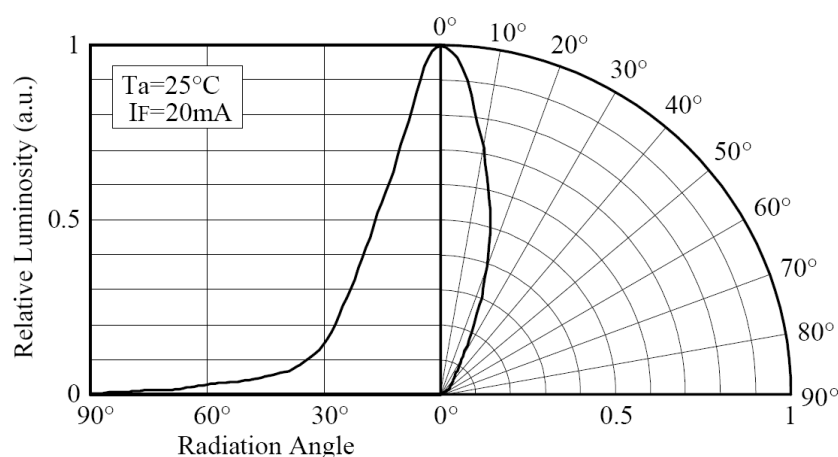


Gráfico da curva fotométrica do LED de cor azul de 1160 mcd¹⁰ típico, com lente moldada em ângulo de 30° e 3mm de diâmetro. Referência do produto: NSPB310A

Figura 14 – Curva Fotométrica LED 30°

[Fonte: http://www.nichia.co.jp/specification/led_lamp/NSPB310A-E.pdf, Pág. 6]

⁹ Lumen – Unidade do sistema internacional de unidades que representa o fluxo luminoso. 1 lumen = 1 candela x ângulo em esferorradiano.

¹⁰ (mcd) - millicandela = 10^{-3} candela, onde candela é unidade do sistema internacional de unidades que representa a intensidade luminosa.

O gráfico (figura 14) em questão traz mais informações sobre a intensidade luminosa do LED. Do seu lado direito temos sua superfície fotométrica, onde diferentemente do gráfico da lâmpada incandescente, o LED está imaginariamente na porção inferior situado na posição 0° e direcionando seu feixe luminoso na direção superior do gráfico. Podemos analisar pelo gráfico que teoricamente um observador situado na posição 60° em relação ao eixo do LED não verá sua emissão luminosa.

No lado esquerdo do gráfico temos uma curva de análise da luminosidade relativa à posição do observador, onde a 90° do eixo a iluminação captada pelo observador é nula e a 0° do eixo a iluminação é máxima.

Devemos observar que simplesmente com a mudança do formato da lente de epoxy do LED podemos mudar completamente o formato da curva fotométrica. Como exemplo podemos apresentar o gráfico abaixo (figura 15) que se refere a outro LED da mesma cor azul e aproximadamente o mesmo valor típico de intensidade luminosa, porém com uma lente forjada com ângulo de 15° (*datasheets* em anexo).

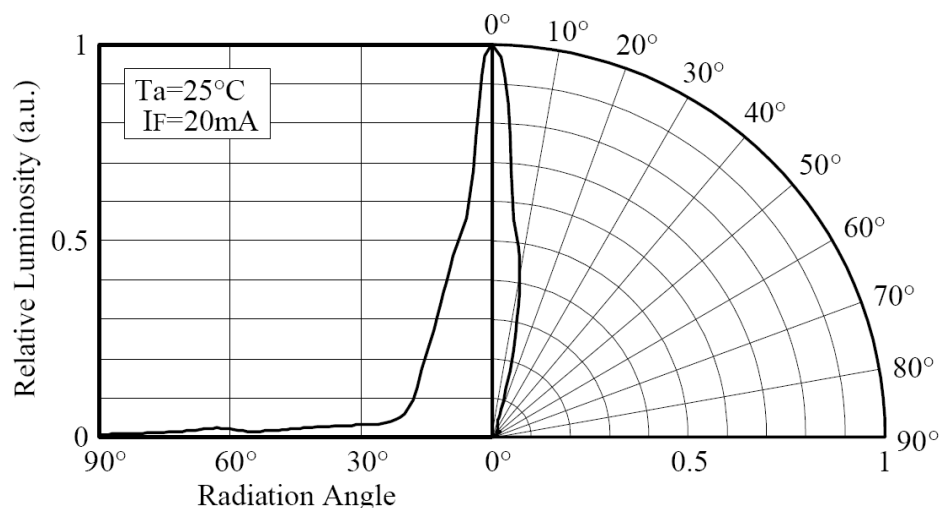
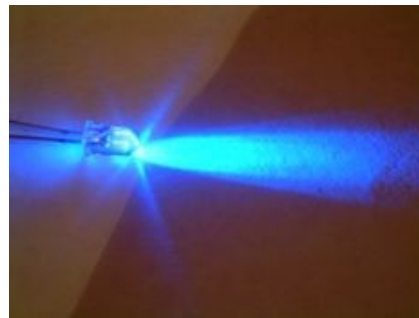


Gráfico da curva fotométrica do LED de cor azul de 2300 mcd típico, com lente moldada em ângulo de 15° e 3mm de diâmetro. Referência do produto: NSPB300A

Figura 15 – Curva Fotométrica LED 15°

[Fonte: http://www.nichia.co.jp/specification/led_lamp/NSPB300A-E.pdf, Pág. 6]

A título de ilustração, as imagens da figura 16, apresentam uma iluminação de LED azul (a) e uma aproximação do núcleo do LED com sua camada epoxy protetora e direcionadora do foco (b).



(a)



(b)

Figura 16 – LED emitindo e Detalhe
[Fonte: Benutzer, C. Pelant (10. Apr 2005)]

CAPÍTULO 3 – SEMICONDUTORES

3.1 - O Semicondutor

Semicondutor é um dos termos mais comuns no mundo tecnológico. Definindo de forma simples, semicondutor é um elemento como o silício e um composto como o sulfite que unidos permitem a passagem e controlam o fluxo de eletricidade. Pode ser alocado entre os bons condutores como o cobre e os maus condutores como o vidro. Desta forma, podemos considerá-lo como um condutor mediano [IEEE VIRTUAL MUSEUM, 2007].

Os semicondutores têm uma estrutura atômica singular que permitem que sua condutividade seja controlada pela estimulação de corrente elétrica, campos magnéticos ou até mesmo a luz. Com isto podemos construir dispositivos que podem amplificar, selecionar, converter luz do sol em eletricidade, ou até mesmo produzir luz a partir da eletricidade [IEEE VIRTUAL MUSEUM, 2007].

Em meados de 1920 os semicondutores despontaram como grande promessa tecnológica, mas somente durante a década de 40 com a utilização da teoria atômica e quântica é que foi desenvolvida a física do estado sólido que investiga as propriedades, estruturas e comportamento elétrico dos semicondutores. [MARQUES, 2001].

Para entendermos melhor o que ocorre eletricamente em um semicondutor precisamos entender primeiro como se comporta o elétron em sua estrutura atômica.

Em um átomo os elétrons giram ao redor do núcleo em órbitas bem definidas, como representado na figura 17.

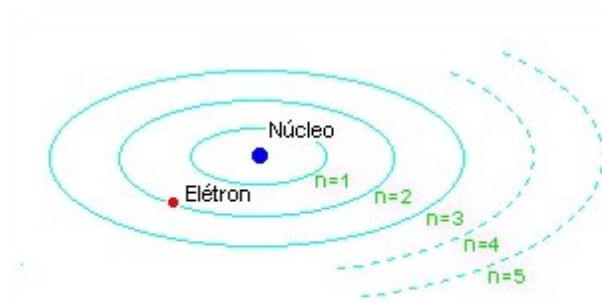


Figura 17 – Órbitas eletrônicas.
 [Fonte: <http://www.ufsm.br/gef/Fmod19.htm>]

Estas órbitas ou níveis são representados como K, L, M, N, O, P e Q, onde K representa o nível mais próximo do núcleo e Q o mais distante.

Pelo terceiro postulado de Bohr “...O elétron irradia, quando salta de um estado estacionário para outro mais interno, sendo a energia irradiada dada por

$$E = hf = E_i - E_f \quad (\text{Equação 3})$$

onde h é a constante de Planck ($6.63 \times 10^{-34} \text{ J.s} = 4.14 \times 10^{-15} \text{ eV.s}$), f é a frequência da radiação emitida, E_i e E_f são energias dos estados inicial e final.” [SANTOS, 2007]. Desta forma, podemos concluir que o elétron situado em uma camada mais externa tem mais energia que um elétron situado em uma camada mais interna.

Para continuarmos a apresentar o conceito do que é um semicondutor precisamos ainda dizer que “...um elétron precisa estar a uma determinada distância (órbita) do núcleo e com uma determinada velocidade (energia cinética) para que a força centrífuga F_c sobre ele, orientada radialmente e para fora do átomo, equilibre-se com a força eletrostática (F_e) entre ele e o núcleo, orientada radialmente em direção ao núcleo (atração eletrostática), tornando-o estável...” [MARQUES, 2001], desta forma, para haver equilíbrio nas órbitas devem coexistir um número máximo de elétrons por órbita.

O Engº Hilton Andrade de Mello define a tabela 2 para o número máximo de elétrons por órbita.

Tabela 2 – Número de elétrons por órbita

<i>Camada</i>	<i>n</i>	<i>Número máximo de elétrons</i>
K	1	2
L	2	8
M	3	18
N	4	32
O	5	50
P	6	72
Q	7	98

Fonte: [MELLO, 1980]

Como já dito, anteriormente, o elétron pode se libertar do átomo se for bombardeado por fonte externa de luz, calor ou magnetismo. Os elétrons mais prováveis de saírem de suas órbitas estão localizados na última órbita. A esta última órbita chamamos de banda de valência ou mesmo órbita de valência. Desta forma, uma pequena quantidade de energia pode transformar o elétron em elétron livre.

Mas os elétrons livres somente podem trafegar pelo material na chamada banda de condução que seria uma camada que ligaria vários átomos do mesmo material. Sobre a ação de um campo elétrico estes elétrons livres que se movem na banda de condução formam a corrente elétrica (i).

Como já explicado, anteriormente, a união de três fatores (energia cinética, força centrífuga e força eletrostática) faz com que o elétron sempre esteja localizado em uma órbita e como a carga do elétron é fixa ($1e = 1,6 \times 10^{-19} \text{ C}$) e sua massa também ($9,11 \times 10^{-28} \text{ g}$) estas órbitas são bem definidas ao redor do núcleo do átomo, ficando determinadas regiões impossibilitadas de receber elétrons. A estas regiões, que também podem ser chamadas de órbitas, chamamos de banda proibida.

Com a definição dos conceitos de banda proibida, banda de valência, banda de condução e elétron livre podemos definir os materiais como condutores, semicondutores e isolantes.

Isolantes: Aqueles materiais que entre sua banda de valência e sua banda de condução apresentam uma banda proibida muito extensa são chamados de isolantes. Para que o elétron se torne livre, o mesmo deve ser bombardeado por uma quantidade muito grande de energia.

Condutores: Aqueles materiais que entre sua banda de valência e sua banda de condução apresentam uma banda proibida bem pequena são chamados de condutores. Para que o elétron se torne livre o mesmo deve ser bombardeado por pequenas quantidades de energia. Até mesmo a temperatura ambiente pode conduzir elétrons de sua banda de valência para a banda de condução. Normalmente, características intrínsecas aos metais.

Semicondutores: Aqueles materiais que entre sua banda de valência e sua banda de condução apresentam uma banda proibida pouco extensa (entre os casos anteriores) são chamados de semicondutores. Para que o elétron se torne livre o mesmo deve ser bombardeado por pequenas quantidades de energia. [MARQUES, 2001]

Vários fatores são causadores dos tamanhos diferentes das bandas, como por exemplo a composição química, o tipo de ligação entre os átomos (covalente, iônica ou metálica), ou sua organização (cristalina ou amorfa), porém, este estudo não é de fundamental importância para este trabalho.

3.1.1 - Semicondutores intrínsecos

Intrínseco no dicionário é aquele ...”que está inseparavelmente ligado a uma pessoa ou coisa; inerente; peculiar” [FERREIRA, 1999]

Podemos dizer então que um semicondutor intrínseco é um semicondutor puro. Um semicondutor puro extrapolamos estatisticamente que seja um semicondutor que, na prática, tenha 1 (uma) impureza para cada 10^9 (dez elevado a nona potência) partes do material, sendo esta impureza desconsiderada do comportamento do material. Estas impurezas são aquelas que não puderam ser eliminadas no processo de purificação do material.

Uma conclusão importante é que “...em um semicondutor intrínseco, o número de buracos é sempre igual ao número de elétrons, uma vez que, ao ser uma ligação covalente quebrada, surgem simultaneamente um buraco e um elétron, e, quando os buracos e os elétrons se recombinam, ambos são eliminados no processo¹¹.” [MELLO, 1980]

3.1.2 - Impurezas Doadoras

Quando ao semicondutor intrínseco é adicionado um átomo que em sua camada de valência possui cinco elétrons, como por exemplo átomos de arsênio (As), antimônio (Sb) e fósforo (P), o mesmo se liga à estrutura do semicondutor através de quatro ligações covalentes, sobrando então um elétron livre. Este átomo, ou o conjunto de átomos, chamamos de impureza pentavalente ou impureza doadora.

“Como os elétrons livres são cargas elétricas negativas, este semicondutor é chamado tipo N.” [MARQUES, 2001].

¹¹ “Logicamente, quando há uma recombinação, a energia dos buracos e elétrons eliminados é liberada de alguma forma: algumas vezes, fótons são emitidos e, outras, a energia liberada é adicionada à energia de vibração da rede (emissão de fônons (subpartícula atômica – Energia Vibracional)).” [MELLO, 1980]

3.1.3 - Impurezas Receptoras

De forma semelhante, quando ao semicondutor intrínseco é adicionado um átomo que em sua camada de valência possui três elétrons, como por exemplo átomos de alumínio (Al), boro (B), e gálio (Ga), o mesmo se liga à estrutura do semicondutor, através de três ligações covalentes, deixando o semicondutor com excesso de lacunas (ou buracos), havendo então a carência de elétrons. Este átomo, ou o conjunto de átomos, chamamos de impureza trivalente ou impureza aceitadora.

Vale salientar que estas ligações covalentes, para serem quebradas, precisam de uma energia determinada. No caso do Germânio são necessários 0,7 eV para quebrar a ligação covalente e, possivelmente, formando-se buracos na estrutura. No caso do silício são necessários 1,11 eV. Energias estas que podem ser obtidas, por exemplo, apenas aquecendo o material. [MELLO, 1980]

“Como as lacunas podem ser consideradas cargas elétricas positivas, este semicondutor é chamado tipo P.” [MARQUES, 2001]

Chamamos de dopagem o processo de adição de impurezas ao semicondutor intrínseco.

“No caso dos semicondutores III-V (GaAs e InP), não existe um processo de dopagem propriamente dito. Os semicondutores tipo N e P, no caso do GaAs, são obtidos através do aumento da dose de Ga (para tipo N) e de As (para tipo P).” [MARQUES, 2001]

3.1.4 - Intervalo (Gap)

Para que o elétron localizado na banda de valência salte para a banda de condução é necessário que o mesmo receba uma quantidade de energia. Podemos representar esta energia por E_g , onde o índice g, proveniente da palavra inglesa *gap*, significa intervalo. [REZENDE, 2004] .

Uma outra forma de caracterizar um material como semicondutor é dizer que os mesmos são materiais isolantes, que em seu estado inicial, apresentam um E_g pequeno (de aproximadamente 1eV) à temperatura ambiente. Desta forma, o semicondutor apresenta uma condução significativa, que ainda assim é muito menor que nos metais.

Usando a energia de *gap* podemos exemplificar com a tabela 3 alguns materiais como condutores ou isolantes.

Tabela 3 – Energia necessária para salto

<i>Material</i>	<i>E_g</i>	<i>Característica do Material</i>
Silício	1,1 eV	Semicondutor
Diamante	5 eV	Isolante
SiO ₂ (óxido de silício)	8 eV	Isolante
Germânio	0,67 eV	Semicondutor
InGaAs	0,4~1,4 eV	Semicondutor

Fonte: [REZENDE, 2004], modificado pelo autor.

3.1.5 - Camada de depleção

Para os estudos dos semicondutores e conseqüentemente dos LEDs precisamos ainda definir o que é a camada de depleção.

Também chamada de região de transição, a camada de depleção é criada quando existe a união entre dois cristais, um do tipo N e outro do Tipo P, e é formada nas proximidades da junção. A camada de depleção contém os íons positivos e negativos deixados no material pela adição de impurezas. Estas cargas iônicas na região de depleção formam a chamada barreira de potencial que aumenta e diminui conforme exista a aplicação de uma diferença de

potencial (DDP) nos extremos opostos à junção. Se a DDP for aplicada por polarização direta (positivo da fonte ligado ao semiconductor tipo P e negativo da fonte ligado ao semiconductor tipo N) existirá a corrente elétrica no circuito. Se a DDP for aplicada por polarização inversa (positivo da fonte ligado ao semiconductor tipo N e negativo da fonte ligado ao semiconductor tipo P. Conforme figura 18) não existirá a corrente elétrica no circuito, pois haverá o aumento da barreira de potencial impedindo a passagem dos elétrons.

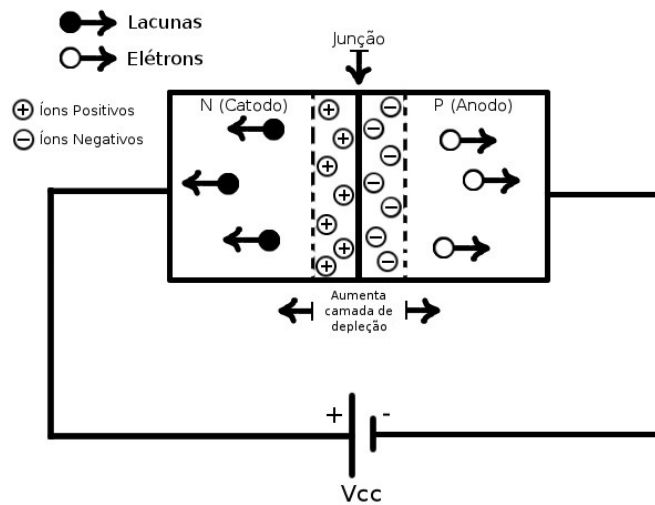


Figura 18 – Polarização Reversa
[Fonte: O Autor]

Este tipo de formação que une semicondutores tipo P e tipo N chamamos de Diodo e um de seus formatos mais comuns é apresentado na figura 19. As marcações em prata e preto representam o catodo (K) que é a porção externa do semiconductor do tipo N.

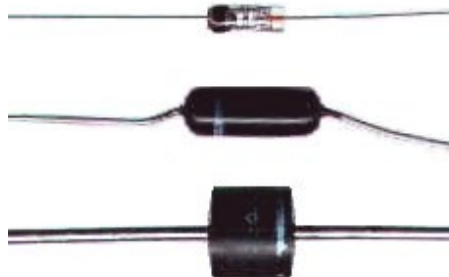


Figura 19 – Diodos axiais.
[Fonte: http://profs.ccems.pt/PaulaFrota/indice_9_ano.htm]

3.2 - O diodo emissor de luz (LED)

Um tipo especial de diodo é o LED – **L**ight **E**mitting **D**iode que é muito utilizado na indústria de equipamentos eletrônicos como indicador e, mais atualmente, na iluminação de semáforos de trânsito e para a formação de imagens digitais em painéis luminosos.

Da mesma forma que nos diodos, os LEDs são formados por semicondutores p e n, mas enquanto nos diodos comuns, as recombinações entre elétrons e buracos geram fótons e fônons¹², o que gera uma dissipação térmica, no caso dos LEDs há uma preferência de fabricação de LEDs com semicondutores de *gap* direto que não necessitam da emissão de fônons e apresentam um fluxo maior de emissão de fótons.

O processo de conversão de energia elétrica em luz é extremamente eficiente nos LEDs. “Nas lâmpadas incandescentes comuns, apenas 13% da energia elétrica são convertidos em energia luminosa” [REZENDE, 2004], enquanto já existem estudos acadêmicos que atingiram estatisticamente 100% de eficiência na conversão¹³[JABBOUR, 2006].

Sobre o processo de luminescência nos LEDs, ele ocorre quando os elétrons da banda de condução estiverem com uma energia (E_c) muito próxima da energia mínima aceita nesta banda. O elétron então decai para a banda de energia menor (banda de valência) recombina-se com um buraco e emitindo um fóton. A energia deste fóton é pouco maior que a energia necessária para vencer o *gap* (E_g) entre as bandas.

Podemos observar na figura 20 o esquema de construção no núcleo de um LED que opera na faixa do vermelho, “...normalmente o lado p é uma camada fina na parte de cima, feita com dopagem muito menor que no lado n. Isto faz com que a radiação seja produzida

¹² Subpartícula atômica relativa a energia vibracional.

¹³ Excimer-Based White Phosphorescent Organic Light-Emitting Diodes with Nearly 100% Internal Quantum Efficiency - Ghassan Jabbour, Jian Li, Evan Williams, Kirsi Haavisto - Advanced Materials- Vol.: Volume 19, Issue 2, Pages 197-202.

majoritariamente no lado p, próximo da janela de saída, por elétrons injetados do lado n...”[REZENDE, 2004], as camadas $\text{GaAs}_{1-y}\text{P}_y$ e GaAs existem, unicamente, por fazerem parte do processo de construção do *chip*, não afetando diretamente a estrutura do LED.

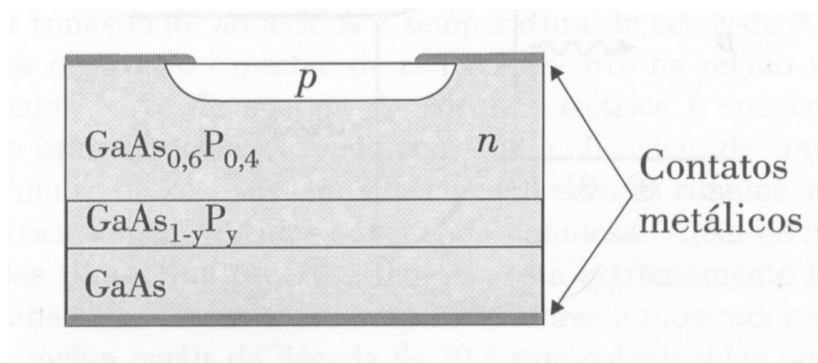


Figura 20 – Estrutura típica do *chip* de um diodo emissor de Luz (LED).

[Fonte: [REZENDE, 2004]]

Como poderíamos imaginar, no chip do LED a emissão luminosa não ocorre em somente uma direção, ela é produzida inicialmente no centro da junção p-n irradiando-se em cones para todas as 6 direções do cubo. No caso do chip do LED composto pela liga AlGaInP apresentada, na figura 21, o semiconductor tipo p (camada muito fina de AlInP) do LED é coberto por um substrato transparente (GaP) que será por onde a maior parte da iluminação será emitida. Quase toda a iluminação gerada na junção que for direcionada para baixo, será absorvida pelo substrato da camada tipo n (GaAs), desta forma observa-se que grande parte da iluminação gerada é perdida pela absorção do material. Mas nem tudo está perdido, os outros 4 cones de luz formados na lateral do *chip* serão parcialmente absorvidos pelo substrato tipo n e parcialmente emitidos lateralmente. Normalmente os fabricantes depositam o *chip* do LED sobre (e dentro) de um cone metálico no intuito de redirecionar estas emissões laterais (já visto na figura 16(b)).

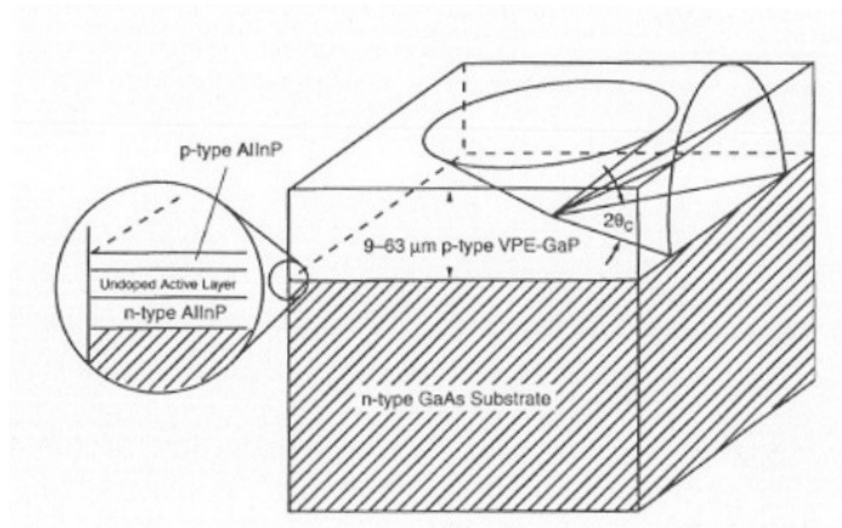


Diagrama simplificado de um *chip* de LED com janela de abertura de GaP.
Os contatos metálicos foram omitidos para facilitar a visualização.

Figura 21 – Formação de cones de luz no chip de LED

[Fonte: [STRINGFELLOW, 1997]]

Devemos lembrar também que toda a radiação emitida pelo centro do chip que atingir a superfície com ângulo superior ao ângulo crítico de reflexão será refletido para seu interior novamente e, posteriormente, absorvido pelo substrato tipo n. Algumas técnicas que envolvem a adição de uma camada de reflexão antes do substrato tipo n, e técnicas como a melhoria da transparência do substrato tipo n, tem sido desenvolvidas [STRINGFELLOW, 1997].

Para determinarmos a angulação dos cones formados no núcleo do chip podemos utilizar a lei de Snell¹⁴ que relaciona os índices de refrações dos dois materiais. Se considerarmos que o material 1 (um) é o ar, então $n_1 = 1$, e o material 2 sendo o GaP então $n_2 \approx 3,4$ (epoxy), teríamos desta forma um ângulo limite do cone de 17,1 graus, fazendo com que o cone tenha abertura de 34,2 graus.

¹⁴ A razão entre o seno do ângulo de incidência (θ_1) e o seno do ângulo de refração (θ_2) é constante e esta constante é igual ao índice de refração relativo n_{21} , para um dado comprimento de onda. Tal que:

$$\frac{\sin \theta_1}{\sin \theta_2} = n_{21} = \frac{n_2}{n_1}, \text{ ou para ângulo limite } \theta_l = \sin^{-1} \left(\frac{n_1}{n_2} \right)$$

Fonte: [GUERRINI, 2007]

Agora imaginemos que após a janela de abertura do *chip* não exista ar e sim uma resina epoxy. Considerando o índice de refração do epoxy igual a 1,5 então teríamos um ângulo limite de 26,2 graus e um cone com 52,4 graus de abertura. Podemos concluir que somente a cobertura do LED por uma camada de epoxy já aumenta sua emissão em quase 50% (cinquenta por cento) [STRINGFELLOW, 1997].

CAPÍTULO 4 – Modulação por Largura de Pulso

4.1 - O que é PWM

Os autores David José de Souza e Nicolás César Lavinia definem PWM como “...uma onda com frequência constante (período fixo) e largura de pulso (duty cycle) variável.” [LAVINIA, 2005].

Podemos dizer que a forma mais usual de representação da modulação por largura de pulso (MLP) é utilizando-se a sigla em inglês "PWM" (Pulse-Width Modulation) e que envolve principalmente a modulação de um sinal usando sua razão cíclica (duty cycle) para transportar qualquer informação sobre um canal de comunicação ou controlar o valor da alimentação entregue a carga [BARR, 2001].

Com o PWM podemos obter uma tensão analógica a partir de um sinal digital. No caso de um microcontrolador e utilizando suas portas de saída meramente digitais, podemos definir dois estados possíveis, 0 e 1.

Na teoria, de forma geral, a tensão média de uma forma de onda é dada pela equação:

$$V_{dc} = \int_0^T \frac{1}{T} V(t) dt \quad (\text{Equação 4})$$

Onde T é o período da forma de onda e V(t) é a função da tensão no tempo.

Para o caso do PWM temos que:

$$V(t) = \begin{cases} V_{pulso} \rightarrow 0 \leq t \leq t_p \\ 0 \rightarrow t_p < t \leq T \end{cases} \quad (\text{Equação 5})$$

Onde: t_p é a duração do pulso em nível lógico 1

V_{pulso} é a tensão de pulso do sinal PWM

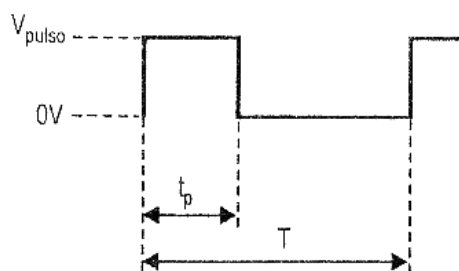


Figura 22 – Representação de um pulso em PWM.

[Fonte: [LAVINIA, 2005]]

Desta forma,

$$V_{dc} = \frac{1}{T} \left(\int_0^{t_p} V_{pulso} dt + \int_{t_p}^T 0 dt \right) \quad (\text{Equação 6})$$

$$V_{dc} = \frac{t_p}{T} V_{pulso} \quad (\text{Equação 7})$$

Sendo assim, temos que a razão entre a largura de pulso e o período da forma de onda recebe o nome de *duty cycle*, ou em português, ciclo ativo. O pulso da onda PWM apresenta tensão fixa, porém o valor médio da tensão desta forma de onda varia em função do *duty cycle*. A tensão média de onda pode variar entre 0 e V_{pulso} (figura 22) [LAVINIA, 2005]. No caso de utilização de uma porta digital de saída de microcontrolador, a variação será de V_{ss} a V_{dd} , ou seja, de 0 a 5V.

4.1.1 - Filtro DC

Não é o caso deste projeto, mas se quisermos utilizar um nível de tensão contínua precisaríamos adicionar à saída da porta do microcontrolador que está programada para a saída de sinal PWM um filtro que passa baixa frequência de corte menor que a própria frequência do PWM.

Para calcular o filtro (F_c) utilizaríamos a fórmula abaixo (figura 23):

$$F_c = \frac{1}{2\pi RC} \quad (\text{Equação 8})$$

Adotar:

$$F_c \leq \frac{f_{PWM}}{10} \quad (\text{Equação 9})$$

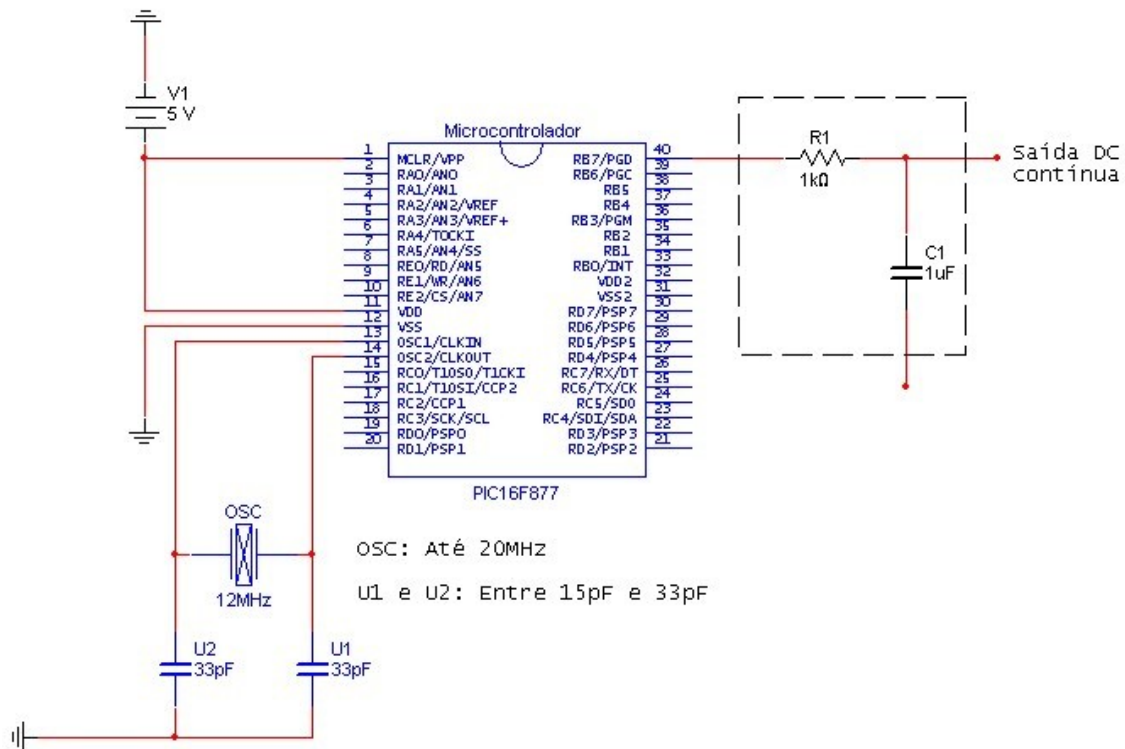


Diagrama representando um possível filtro de PWM para DC, considerando que o microcontrolador esteja programado para emitir pulso PWM na porta RB7.

Figura 23 – Filtro para PWM atingir nível DC

[Fonte: O autor]

4.2 – Comparando PWM com variação de tensão em LEDs

Conforme uma das maiores fabricantes de LEDs de alto brilho do mundo, a japonesa *Nichia*, há basicamente duas formas de controlar a intensidade luminosa de um LED.

A primeira delas seria variando a corrente (DC) de alimentação do LED. Conforme podemos observar na figura 24.

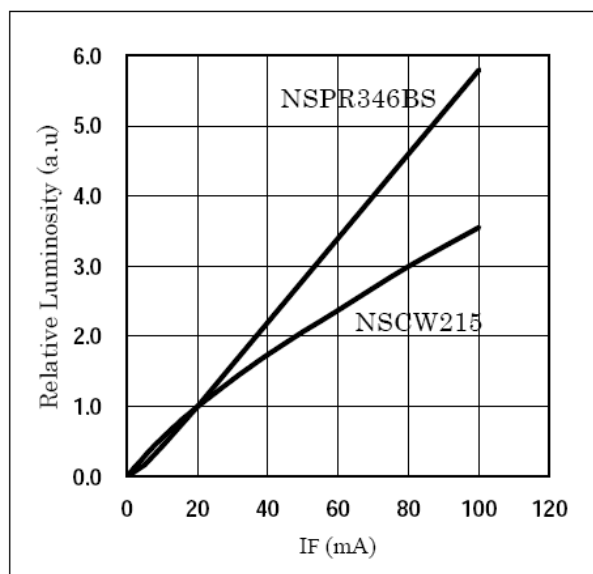


Figura 24 – Variando Tensão do LED Nichia.

Fonte: [NICHIA CORPORATION, 2004]

À medida que aumentamos a corrente também aumentamos a intensidade luminosa dos LEDs. O gráfico acima apresenta dois tipos de LEDs diferentes. O NSCW215 e NSPR346BS que emitem luz branca e vermelha, respectivamente.

Observa-se então que para diferentes cores de LEDs a variação de corrente representa uma variação desproporcional de intensidade luminosa.

No caso específico deste trabalho significaria fazer um controle individual de cada cor, e dependendo da cor, fazer uma correção da curva de luminosidade o que aumentaria o nível de complexidade do projeto.

A segunda forma e mais aconselhada pela própria fabricante seria a utilização da técnica de PWM, pois como a corrente é constante não haveria mudança na cor [NICHIA CORPORATION, 2004].

CAPÍTULO 5 – PIC

Criado pela empresa Microchip os microcontroladores PIC, cujo acrônimo significa *Programmable Integrated Circuits*, juntamente com seus concorrentes diretos os 8051¹⁵ da Intel, dominam o mercado mundial de microcontroladores.

Basicamente as duas empresas têm produtos equivalentes em características, mais para o mercado de Brasília o que é mais encontrado no comércio local são os microcontroladores da família PIC. Como um dos principais objetivos deste projeto é minimizar o custo de produção, foi escolhido trabalhar com os microcontroladores da família PIC, optando por um modelo que apresentasse o maior número de portas de entrada e saída disponíveis.

Mais adiante, apresentaremos técnicas para a utilização do microcontrolador PIC, algumas foram utilizadas para a criação deste projeto.

5.1 - OSCILADORES

Tão importante como uma boa alimentação para o PIC é a utilização de uma boa base de ciclos para a utilização de contadores ou a sincronização de vários PICs no mesmo circuito.

A obtenção de um oscilador para o PIC pode ser implementada de diversas formas, onde as principais variáveis são o custo de implementação e a confiabilidade do pulso. Basicamente, apresentaremos 4 formas de se implementar um oscilador dos quais escolheremos um que será demonstrado neste projeto.

5.1.1 - Oscilador do tipo RC (Resistor-Capacitor)

Sendo o oscilador mais barato do mercado também é de grande versatilidade, pois somente utiliza uma porta do microcontrolador (OSC1). Sua precisão é ruim pois apresenta grande variação em condições diferenciadas de temperatura, tolerância de componentes e variação de tensão.

Para a maioria dos microcontroladores PIC, a utilização dos osciladores do tipo RC não são vantajosas, pois com a utilização dos osciladores internos obtemos um pulso mais regular (figura 25).

¹⁵ Família de Microcontroladores

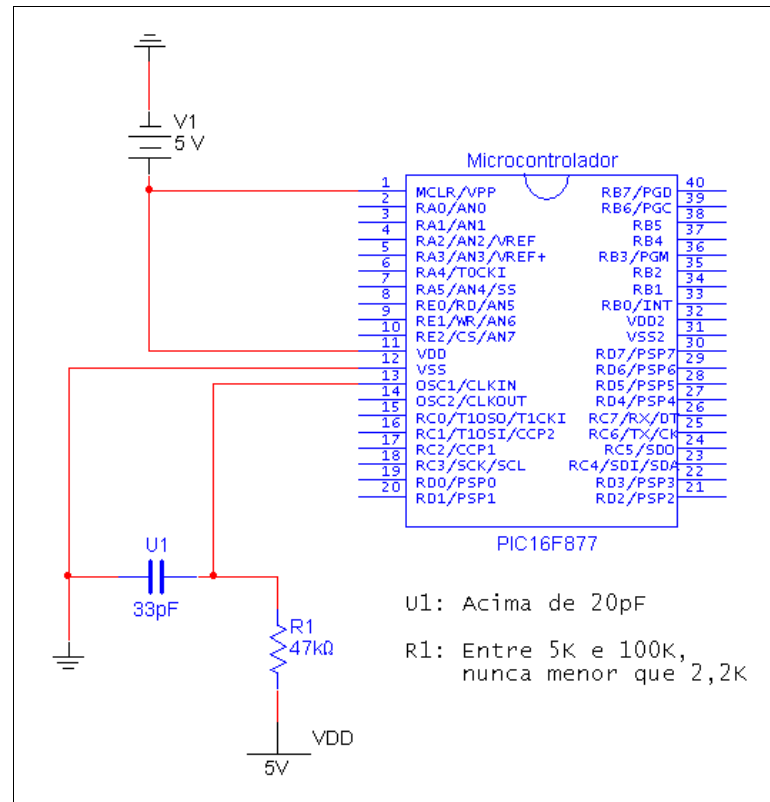


Figura 25 – Oscilador do tipo RC

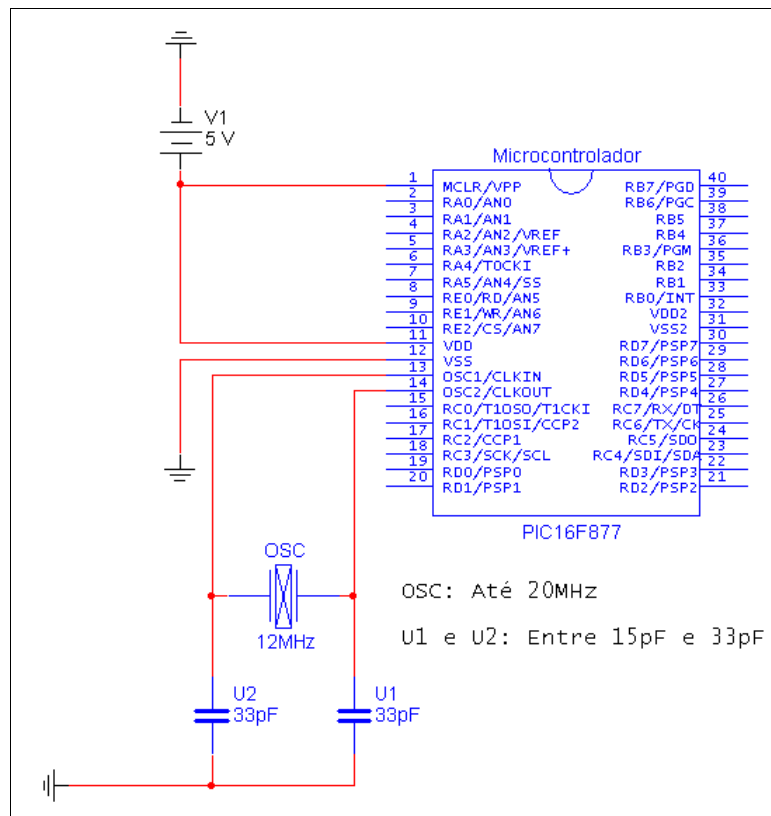
[Fonte: o autor]

5.1.2 - Oscilador do tipo Ressorador

Também conhecido como ressonador, o ressonador cerâmico é uma opção não tão barata quanto o oscilador RC, mas é bem mais preciso e estável. Uma desvantagem em relação ao RC é que o ressonador utiliza duas portas de saída/entrada do microcontrolador.

No mercado encontramos basicamente dois tipos de ressoadores, um com dois pinos e outro com três pinos.

Para a implementação do ressonador de dois pinos faz-se necessária a utilização de dois capacitores cerâmicos (conforme a figura 26) e podendo variar entre 15pF e 33pF. Uma das condições básicas para o funcionamento é que os dois capacitores devem ser iguais e deve ser observado que, quanto maior a frequência maior deve ser o capacitor. A utilização dos capacitores se faz necessária para melhorar a estabilidade do sistema [SOUZA, 2005].



Com o ressoador de três pinos não se faz necessária a utilização de capacitores, conforme figura 27 sugerida abaixo [SOUZA, 2005].

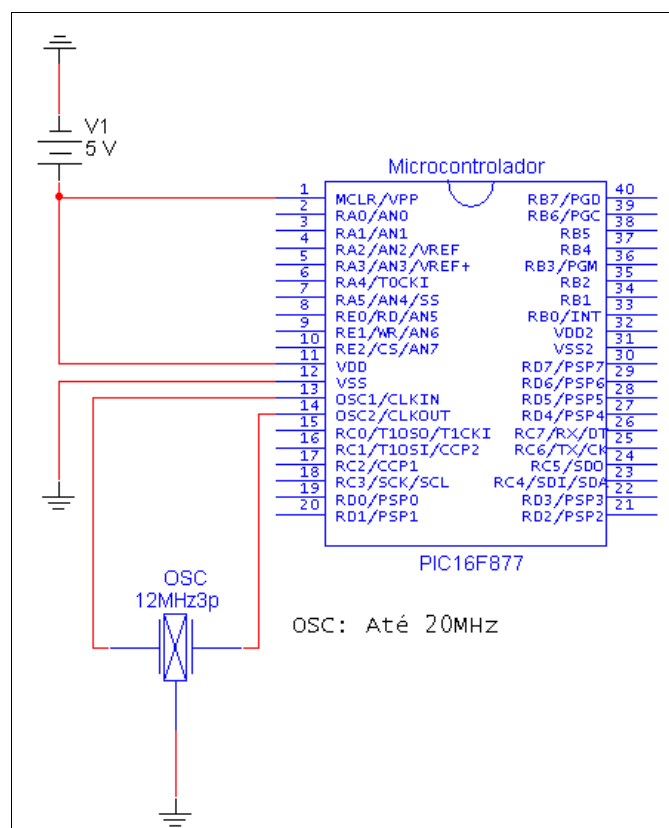


Figura 27 – Ressorador de 3 Pinos

[Fonte: o autor]

5.1.3 - Oscilador do tipo Cristal

Os mais precisos osciladores são os de cristal de quartzo, assim sendo também são os mais caros. A idéia de seu funcionamento é baseada no princípio piezelétrico, onde ao aplicar uma tensão CA através de um cristal, eles vibram na frequência da tensão aplicada. Inversamente, se você forçar uma vibração mecânica, eles geram uma tensão CA.

Alguns cristais na natureza apresentam o efeito piezelétrico. As principais substâncias que produzem esse efeito piezelétrico são o Quartzo, os sais de Rochelle e a Turmalina.

“Os sais de Rochelle têm a maior atividade piezelétrica; para uma dada tensão CA, eles vibram mais do que o quartzo e a turmalina. Mecanicamente, são os mais fracos. Eles se quebram facilmente. Os sais de Rochelle têm sido utilizados para fazer microfones, toca-discos, cabeçotes e alto-falantes. A turmalina mostra a menor atividade piezelétrica, mas é a mais forte dos três. É também a mais cara. Ocasionalmente é utilizada em frequências muito altas. O quartzo constitui-se como um intermediário entre a atividade piezelétrica dos sais de Rochelle e a rigidez da turmalina. Por ele ser barato é facilmente encontrado na natureza, o quartzo é amplamente usado em osciladores de RF¹⁶ e filtros” [LUCAS, 2006].

Os cristais de quartzo somente devem ser utilizados em circuitos, quando faz-se necessária uma precisão maior do sistema, garantindo-se assim uma melhor relação custo-benefício.

¹⁶ Abreviação para Rádio Frequência

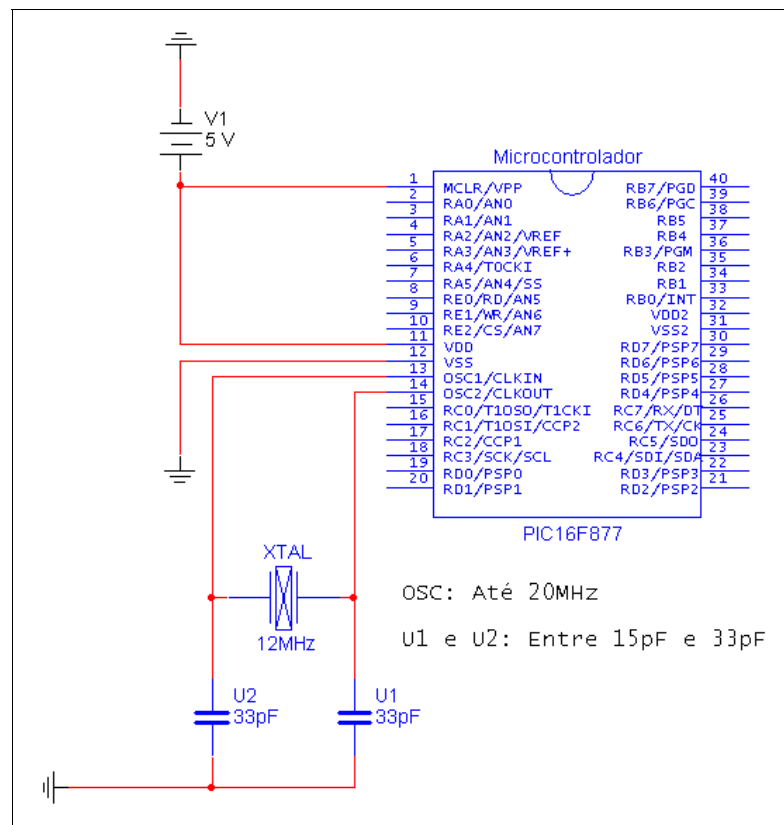


Figura 28 – Cristal de Quartzo

[Fonte: o autor]

No caso específico dos microcontroladores PIC devemos utilizar cristais até no máximo 20MHz, conforme determina o fabricante (figura 28). A implementação do cristal no circuito do PIC segue a mesma estrutura do ressoador de dois pinos já descrita anteriormente. Da mesma forma que os ressoadores, os cristais utilizam dois pinos do microcontrolador.

5.1.4 - Oscilador Interno ao PIC

Alguns modelos de microcontroladores da família PIC possuem osciladores já embutidos em seu circuito interno. Normalmente, estes osciladores internos são equivalentes a um oscilador do tipo RC, já apresentado, e trabalham com frequência de 4MHz (típico), e da mesma forma que os osciladores RC apresentam uma grande variação em sua precisão, normalmente causada por variações de temperatura ou tensão. Esta variação normalmente fica entre 1% e 5%.

Utilizar oscilador interno em determinados casos representa uma vantagem de implementação, pois não será utilizada nenhuma porta de entrada/saída do microcontrolador.

5.1.5 - Outros Osciladores (Circuitos de Oscilação)

Além da utilização de osciladores como cristais ou ressoadores, podemos implementar circuitos que seriam responsáveis pela geração da função degrau correspondente. Em determinadas situações estes circuitos se tornam mais baratos do que a utilização de cristais, pois com eles podemos garantir a sincronia de diversos microcontroladores no mesmo circuito impresso. Em alguns casos podem ser utilizadas implementações híbridas, como por exemplo a utilização de cristais e circuitos osciladores.

CAPÍTULO 6 – MONTAGEM

Neste capítulo apresentaremos algumas técnicas para a montagem de um *display* utilizando LEDs, e nos aprofundaremos em uma delas.

Os LEDs não funcionam sozinhos, precisam de outros componentes eletrônicos para controle de seu brilho e outras características. Uma solução para este problema é a adição ao circuito de componentes analógicos. Outra abordagem é a adição de componentes digitais simulando circuitos analógicos.

Não custa reafirmar que um dos principais objetivos deste trabalho é a eficiência energética do circuito e conforme apresentado na introdução deste projeto podemos destacar a escalabilidade de suas características de controle e a redução de tamanho físico como objetivos secundários. Veremos a seguir que a melhor abordagem é a que maximiza o estado sólido¹⁷ do circuito.

6.1 - O Circuito Analógico

Para se criar a tríade RGB que possibilitará a apresentação de um ponto de luz que emitirá várias cores dentro do espectro visível podemos dividir o trabalho em algumas etapas:

Como será apresentado, a terceira etapa não será implementada, pois conforme concluído na análise da segunda etapa e na conclusão geral da análise, a mesma não será viável, porém será implementada na terceira etapa do circuito digital.

- 1 – Variar a intensidade de brilho de uma das cores que compõem o RGB
- 2 – Ao juntar as três cores que compõem a tríade conseguir apresentar a cor branca

¹⁷ Física do estado sólido – disciplina fundamental que trata do estudo e aprimoramento dos semicondutores.

6.1.1 - Primeira etapa

Desta forma seguiremos com a implementação da primeira fase para o circuito analógico. Para variar o brilho de um LED podemos aplicar dois princípios.

- 1 – Variar a tensão aplicada em seus terminais de contato.
- 2 – Alimentá-lo sempre com a mesma tensão e variar rapidamente os intervalos em que ela será máxima ou mínima (PWM).

Para simplificar o circuito, podemos utilizar circuitos integrados com amplificadores operacionais para controlar a tensão aplicada ao LED e que, inicialmente, optamos por um LED azul de alto brilho.

O circuito integrado que utilizamos para os amplificadores operacionais é o LM324 que possui 14 (quatorze) pinos de contatos e 4 (quatro) amplificadores operacionais, dos quais somente utilizaremos 2 (dois).

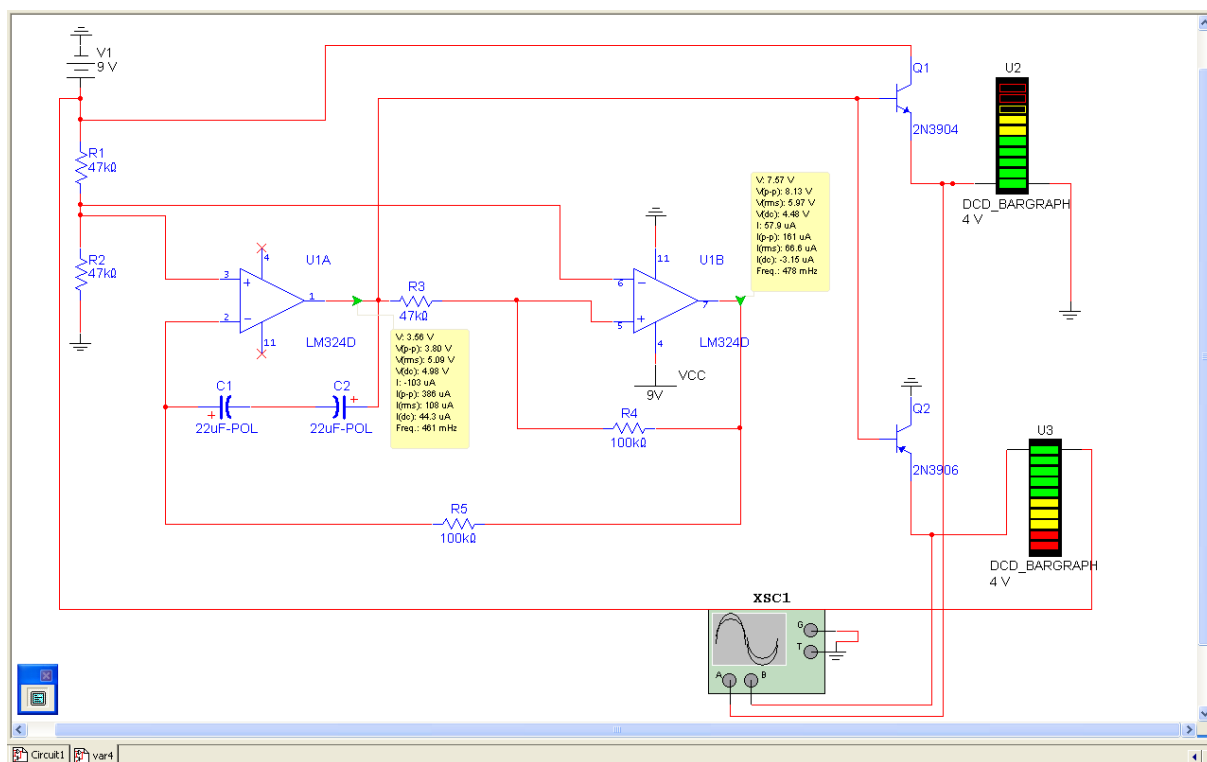


Figura 29 – Circuito para variação de brilho do LED

[Fonte: o autor]

O circuito apresentado na figura 29 foi criado com o simulador Electronic Workbench Multsim 9 e simula a implementação do circuito que varia alternadamente a intensidade de 2 (dois) LEDs.

Este circuito foi baseado no circuito apresentado no site de cooperação técnica criado pelo Sr. Bill Bowden [BOWDEN, 2006] e modificado pelo autor.

Desta forma conseguimos apresentar a primeira parte do projeto com circuito analógico, conforme a figura 30.

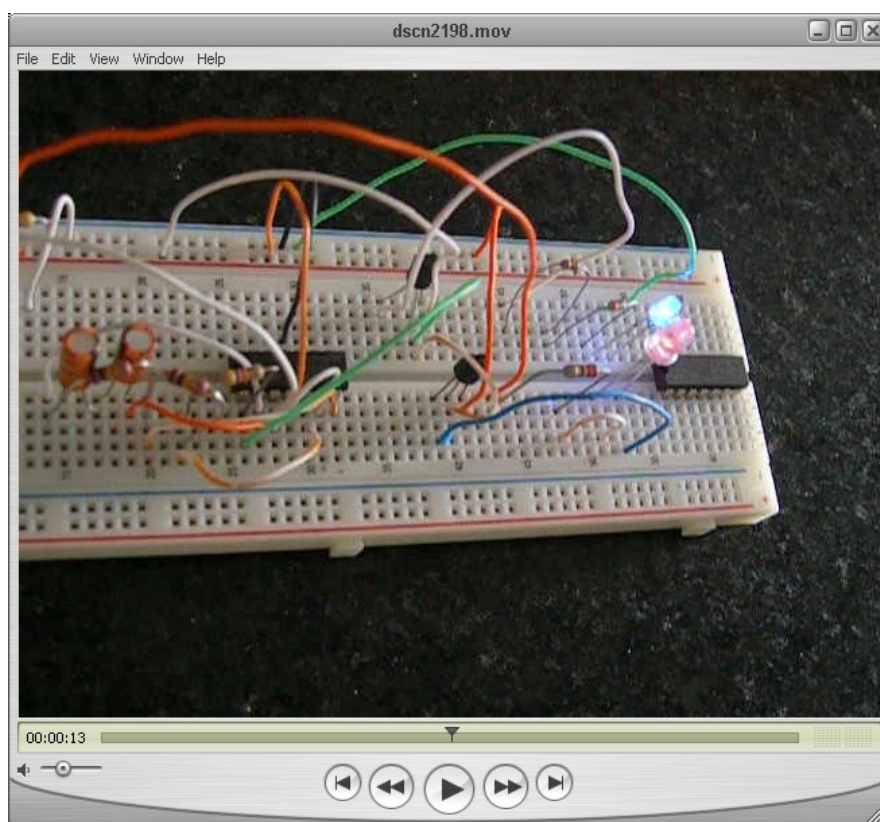


Figura 30 – Implementação do Circuito com AMPOP
[Fonte: o autor]

6.1.1.1 - Conclusões da primeira etapa

Na parte simulada verificou-se que existe uma diferença entre as tensões aplicadas aos transistores de acoplamento dos LEDs, conforme análise do osciloscópio do software utilizado (figura 31). Desta forma, o brilho do LED que está aumentando não se iguala ao do LED que está diminuindo. As duas linhas em vermelho representam cada uma um canal de análise do osciloscópio virtual. Podemos perceber que em um vale qualquer do sinal representado pelo gráfico os dois LEDs estão apagados, porém com tensões de alimentação diferentes o que na prática representará que um estará apagado e outro fracamente aceso. Para o correto funcionamento seria necessário que as duas curvas estivesse sobrepostas.

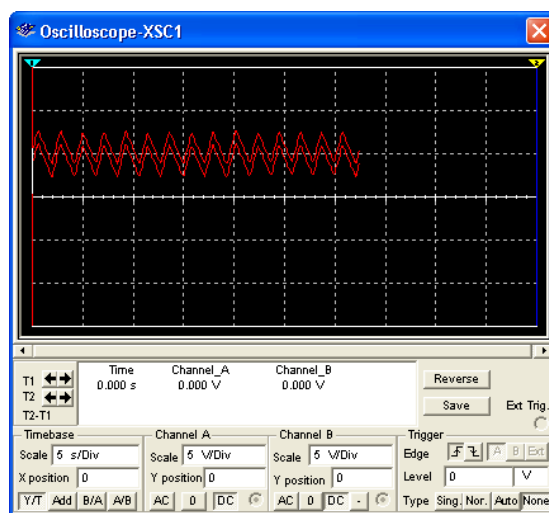


Figura 31 – Análise saída AMPOP pelo osciloscópio.

[Fonte: o autor]

Como já esperado, foi constatado o mesmo resultado com a implementação elétrica do circuito na *protoboard*. O brilho do LED foi variado lentamente conforme esperado.

6.1.2 - Segunda etapa (procedimento e conclusões)

Na segunda etapa de implementação, identificamos que ao adicionarmos mais dois LEDs com as 2 cores faltantes ao circuito o mesmo apresentava mais brilho em alguns LEDs

do que em outros devido, em parte, pela característica de cada LED. Para conseguirmos regular o brilho de cada cor e apresentar a cor branca poderíamos alterar o valor da resistência de acoplamento do LED, assim, diminuindo linearmente seu brilho.

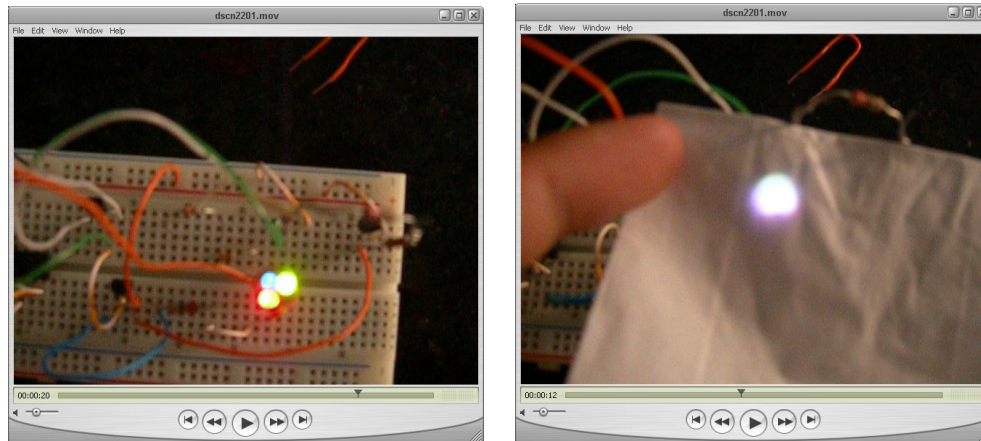


Figura 32 – Análise 3 LEDs analógico.
[Fonte: O autor]

Ao adicionarmos os 3 LEDs ao circuito, (figura 32) a variação de brilho foi aumentada em velocidade ficando mais rápida. Uma solução para esta dificuldade seria regular a resistência de $100K\Omega$ ligada ao pino 5 do LM324 para compensar a velocidade. Quando foi implementada esta alteração, notou-se pouca diferença em relação ao estado anterior.

6.1.3 - Conclusões gerais do circuito analógico.

- 1 – O circuito gasta muita energia, pois existem muitas resistências envolvidas.
- 2 – O circuito utiliza 2 amplificadores operacionais para cada LED, imaginando que cada LED tenha que ter 2 AMPOPs¹⁸ teríamos um total de 54 AMPOPs para a tela final o que aumentaria as dimensões do circuito final e sua dissipação térmica.
- 3 – Para cada LED temos que ter um transistor de acoplamento com a alimentação do circuito.
- 4 – O excesso de componentes no circuito não possibilitará que o mesmo seja reduzido pois teremos muitos pinos e muitos componentes para integrar.
- 5 – Teremos que adicionar capacitores eletrolíticos ao circuito o que aumentaria as dimensões do circuito final e sua dissipação térmica.

¹⁸ Abreviação de Amplificadores Operacionais.

- 6 – Para realizar o controle individual de cada cor teremos que ter uma fonte de sinal externa como um microcontrolador ou um computador o que encareceria muito o circuito.
- 7 – Conforme já apresentado (no capítulo 4, subitem 4.2) os fabricantes de LEDs não aconselham a variação de tensão como técnica aconselhável para variação do brilho.

Pelos pontos aqui abordados, acreditamos que continuar com a construção utilizando amplificadores operacionais seja uma opção não viável. Desta forma, partimos para a implementação em estado sólido¹⁹.

Acreditamos que possa existir uma forma de implementação que possa ser viável e utilize amplificadores operacionais como por exemplo utilizando os circuitos integrados LTC3783, porém seriam necessários maiores esforços e mais tempo para realizar um estudo mais amplo.

6.2 - O Circuito Digital

Ao iniciar o desenvolvimento do projeto digital identificamos a necessidade de pesquisar alguns conceitos em detrimento de algumas perguntas, que ainda estavam sem resposta, como por exemplo:

- 1 – Que intensidade de cada cor da tríade RGB seria necessária para a obtenção das outras 16 cores.
- 2 – Qual o valor correto da resistência que ligaria o LED a seu aterramento (resistência de acoplamento).

Desta forma, antes de continuarmos a apresentar o projeto digital, apresentaremos algumas possíveis soluções para estas questões.

¹⁹ Física do estado sólido – disciplina fundamental que trata do estudo e aprimoramento dos semicondutores.

6.2.1 - O resistor de acoplamento do LED

Da mesma forma que no circuito “analógico” precisamos dimensionar um resistor de acoplamento entre o LED e o circuito, esta necessidade faz-se pelo fato da resistência interna do LED ser muito pequena, o que pode danificar o LED ou o próprio circuito. [REZENDE, 2004]

Segundo o Dr. MALVINO [MALVINO, 1997], para calcularmos a resistência de acoplamento devemos utilizar a lei de Ohm da seguinte forma.

$$R_e = \frac{5V}{20mA} = 250 \Omega \quad (\text{Equação 10})$$

Já estamos considerando a tensão máxima de saída da porta do microcontrolador e corrente ideal de alimentação do LED (conforme *datasheets* em anexo).

Não devemos esquecer de calcular a potência do resistor.

$$P_R = 5V \cdot (20mA)^2 = 0,002 W = 2 mW \quad (\text{Equação 11})$$

Comercialmente encontramos o resistor de 270Ω e $1/8$ de W ($0,125 W$) que irá atender a necessidade do circuito.

6.2.2 - A paleta de cores CGA

Existem diversas paletas de cores para monitores de vídeo (CRT). Com o passar dos anos, a humanidade parou de usar monitores monocromáticos para usar os coloridos, mas para chegar a tecnologia que observamos hoje, muitas etapas foram transpostas.

Um dos primeiros padrões coloridos foi o próprio RGB que era utilizado em monitores de computadores que apresentavam somente 4 cores.

Para o monitor do tipo CGA²⁰ fixo tínhamos 2 (dois) padrões adotados dependendo do fabricante:

1 – Preto, Magenta, Cyan e Cinza claro

2 – Preto, Vermelho, Verde e Marron alaranjado

Foi então que, se adicionando um quarto bit que regulava a intensidade de cada cor conseguiu-se a reprodução de uma paleta de 16 (dezesesseis) cores denominada CGA *full* (ou RGBI). Internamente, as cores no monitor CGA de 16 (dezesesseis) cores são formadas da seguinte forma:

Idealizando que cada cor do sistema RGB deva ser dividida em 4 níveis de intensidade: Desligado (Preto) com 0,00% (zero por cento) de seu brilho máximo, 33,3% (trinta e três por cento) de seu brilho máximo, 66,6% (sessenta e seis por cento) de seu brilho máximo e 100% (cem por cento) de seu brilho máximo, criamos a tabela 4.

Para efeito de representação, cada cor deve ser representada por dois caracteres em hexadecimal sendo 00h – preto e FFh – brilho máximo da cor. Para as intensidades intermediárias teríamos também a tabela 4.

Tabela 4 – Níveis de Brilho em Hexadecimal

<i>Intensidade</i>	<i>%</i>	<i>Hexadecimal</i>	<i>Decimal</i>
Preto	0 %	00h	0
Escuro	33,3 %	55h	85
Brilhante	66,6%	AAh	170
Máxima	100%	FFh	255

[Fonte: O autor]

Sendo assim, podemos observar que os 2 primeiros dígitos em hexadecimal representam a cor vermelha, os 2 dígitos do meio, a cor verde e os 2 (dois) últimos dígitos a cor azul, a forma correta de representar as 16 (dezesesseis) cores segue a figura 33.

²⁰ Acrônimo do inglês *Color Graphics Array*

Paleta CGA 16-cores completa	
0 — Preto #000000	8 — (escuro) Cinza #555555
1 — Azul #0000AA	9 — Azul Claro #5555FF
2 — Verde #00AA00	10 — Verde Claro #55FF55
3 — Cyan #00AAAA	11 — Cyan Claro #55FFFF
4 — Vermelho #AA0000	12 — Vermelho Claro #FF5555
5 — Magenta #AA00AA	13 — Magenta Claro #FF55FF
6 — Marron #AA5500	14 — Amarelo #FFFF55
7 — Branco (Cinza Claro) #AAAAAA	15 — Branco Claro #FFFFFF

Figura 33 – Relação de 16 cores em Hexadecimal.

[Fonte: o autor]

Podemos perceber então que o Verde Claro, por exemplo, é composto por uma intensidade baixa de Vermelho e Azul que deve ser considerada.

6.2.3 - A implementação do circuito digital

Para analisarmos o desempenho e manter a mesma metodologia de trabalho do circuito analógico seguimos as mesmas etapas:

- 1 – Variar a intensidade de brilho de uma das cores que compõem o RGB
- 2 – Ao juntar as três cores que compõem a tríade conseguir apresentar a cor branca
- 3 – Conseguir apresentar as 16 (dezesesseis) cores básicas em 1 (um) ponto.

Em sua fase inicial de implementação, por dificuldade de encontrar no mercado local o microcontrolador da fabricante microchip e modelo PIC 16F877, que seria o microcontrolador ideal para o projeto, foi utilizado o microcontrolador PIC 16F873 do mesmo fabricante.

A dificuldade oriunda de se utilizar o PIC 16F873 deve-se ao fato de que o mesmo somente possui 28 (vinte e oito) pinos o que reduz a quantidade de possíveis LEDs a serem ligados no mesmo (maiores detalhes, consultar *datasheets* dos microcontroladores).

A escolha do PIC 16F877, para finalizar o projeto deve-se ao fato de que na época de sua análise seria o microcontrolador PIC com melhor relação custo-benefício, levando-se em consideração características como: maximização de portas de saída, velocidade interna de processamento, tamanho da memória flash interna e custo do microcontrolador.

A decisão de utilizar microcontroladores da família PIC em detrimento a outros microcontroladores de outros fabricantes, como a Intel (8051), deve-se ao fato de que a Microchip, fabricante do PIC, fornece um conjunto de softwares de edição de código e gravação do microcontrolador gratuitamente, o que baixaria o custo de produção do projeto.

6.2.3.1 - Primeira etapa

Para fazer a ligação do microcontrolador utilizamos uma fonte de energia DC de 15 V e um regulador de tensão LM7805 que converte os 15V originais em 5Vdc com corrente máxima de 1 A. Para filtrar a sujeira da alimentação utilizamos um capacitor eletrolítico de 220 μ F .

Com a fonte de alimentação já fornecendo 5V, ligamos o pino 1 (MCLR negada), pino 8 (Vss) e pino 19 (Vss) ao positivo da fonte de 5V e o pino 20 (Vdd) ao negativo da fonte.

O cristal de 12MHz também foi ligado ao circuito, conforme já explicado no capítulo 5, observando-se a alteração dos pinos de ligação, de acordo com o *datasheet* do PIC.

Utilizando-se a porta RB7 do PIC 16F873 ligamos a mesma a uma resistência de acoplamento, que por sua vez está ligada ao anodo do LED e o catodo do LED ligado no negativo da fonte de 5V.

O microcódigo do microcontrolador PIC 16F873 foi gravado utilizando-se uma gravadora para microcontroladores PIC. Seu código fonte, originalmente foi feito na linguagem C, compilado e linkeditado por programa com esta finalidade. O código de variação da intensidade de brilho do LED encontra-se como apêndice 1 a este trabalho. A figura 34 apresenta o circuito implementado para análise na primeira etapa.

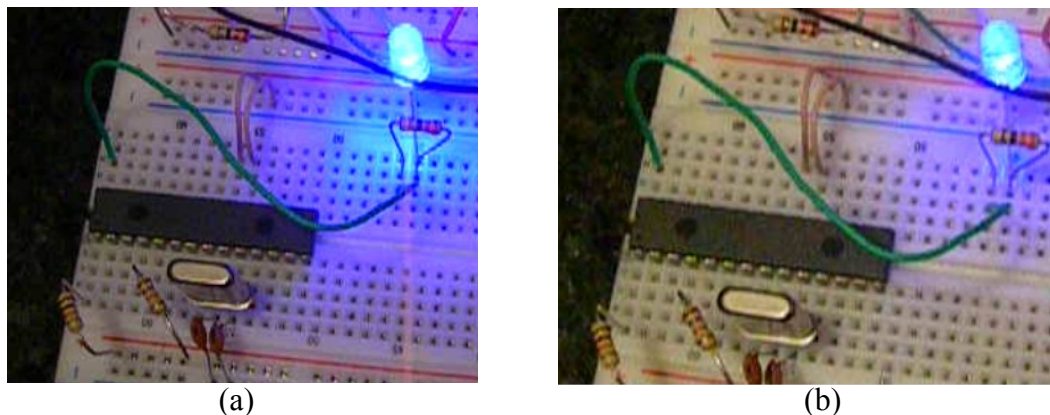


Imagem (a) com brilho mais intenso, Imagem (b) com brilho mais tênue. Não é apresentada na imagem a fonte de alimentação do microcontrolador.

Figura 34 – Variação da intensidade de brilho com PWM

[Fonte: O autor]

6.2.3.1.1 - Conclusões da primeira etapa

Nitidamente percebemos a limpeza do circuito, em relação ao circuito analógico. Somando-se a isso percebemos que o controle de brilho pelo processo de PWM é possível e muito eficiente.

Destacamos porém que, em relação ao circuito analógico, existe uma desvantagem. Quando visto a olho nu o LED, realmente, aparenta ter atenuado sua intensidade, mas quando filmado por outro equipamento percebemos que existe uma cintilação de seu brilho (filmagem

realizada com intuito de perceber o que olho humano não consegue captar). Como este projeto se destina à visão humana esta característica não impede sua continuação.

6.2.3.2 - Segunda etapa

A segunda etapa foi iniciada utilizando-se o mesmo circuito da primeira, adicionando-se o LED verde e o LED vermelho. Cabe salientar que, para se obter a cor branca utilizando-se 3 LEDs precisamos diferenciar o brilho de cada cor, O LED vermelho deve ter brilho próximo de 8000mcd, o LED verde deve ter brilho próximo de 5000mcd e o azul próximo de 2000mcd. Se colocarmos lado a lado LEDs com brilhos iguais certamente o LED azul irá se sobressair. Isto ocorre porque as cores que tendem para o azul apresentam maiores frequências e menores comprimentos de onda, desta forma se dispersando mais sobre o ar e invadindo as outras cores. A estrutura da retina humana comprova isto, pois somente 6% dos cones são destinados à captação da cor azul. Entende-se que a luz azul vai chegar ao olho de qualquer forma, enquanto que a vermelha deve estar direcionada para o olho.

Utilizando-se código fonte similar ao empregado na primeira etapa e adicionando os LEDs verde e vermelho às portas RB6 e RB5 conseguimos emitir luz equivalente a cor branca. Para apresentarmos a cor branca com o observador muito próximo aos LEDs precisamos de algum filtro que possa dissipar a luz de forma a misturar as cores.

Existem diversos estudos relativos à disposição dos LEDs em relação ao observador, para que o branco seja melhor apresentado utilizando não 3, porém 6 ou mais LEDs, para apresentar a cor branca [MORENO, 2005]. Não é objetivo principal deste trabalho apresentar a melhor disposição dos LEDs, tendo em vista que imaginamos que o *display* será utilizado à distância mínima de 3 metros do observador.

Após a obtenção da cor branca, o microcontrolador PIC 16F873 foi substituído no circuito pelo PIC 16F877 que, conforme já comentado será o microcontrolador utilizado no estado final deste projeto. A mudança de código para a troca se constitui basicamente em

alterar as bibliotecas de cabeçalho compiladas e gravadas nos dois microcontroladores. Com relação à pinagem entre os microcontroladores, as trocas devem ocorrer conforme o *datasheet* do fabricante. Lembramos que como os microcontroladores são da mesma família e séries o fabricante se reservou ao direito de publicar somente um *datasheet* que engloba os dois microcontroladores.

6.2.3.2.1 - Conclusões da segunda etapa

A cor branca foi obtida com uma mínima variação do código fonte da etapa anterior, o que reafirma a escalabilidade do circuito. A variação de intensidades e a cor branca foram obtidas, tanto com o microcontrolador PIC 16F873 quanto com o PIC 16F877.

6.2.3.3 - Terceira etapa

O mesmo circuito já implementado para o modelo da segunda etapa será utilizado para a análise da terceira etapa, pois a diferenciação será feita somente pelo código fonte aplicado no microcontrolador.

Conforme descrito, nesta etapa já estamos utilizando o PIC 16F877.

Para a apresentação da 16 cores, iremos utilizar o padrão descrito anteriormente no subitem 6.2.2 que apresenta a paleta de cores CGA. No código fonte utilizaremos a mesma representação hexadecimal para facilitar o entendimento e a legibilidade do código. O circuito está apresentado na figura 35 .

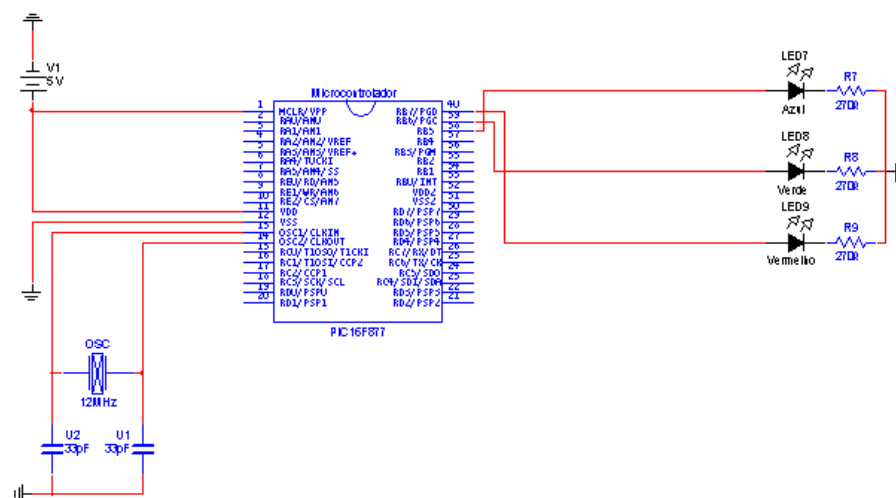


Figura 35 – Circuito para a terceira etapa de análise do circuito digital.

[Fonte: O autor]

6.2.3.3.1 - Conclusões da terceira etapa

As 16 (dezesseis) cores foram apresentadas conforme planejado e o sistema não apresenta diminuição de sua velocidade de processamento em função da quantidade de código acrescida em relação à segunda etapa. Conforme já havia sido planejado, as cores não apresentam definição, à curta distância, sem um filtro visual que dissipe a luz nele incidente.

6.3 – Comparações entre as análises

Para facilitar a compreensão das conclusões obtidas com a análise dos 2 modelos de implementação será apresentada a tabela 5 que apresenta os resultados obtidos. Devemos salientar que as informações apresentadas na tabela são somente as principais conclusões. O detalhamento completo já foi apresentado em tópicos anteriores.

Tabela 5 – Resumo das conclusões de análises dos circuitos.

<i>Objetivo a ser alcançado na etapa de avaliação</i>	<i>Resultado/Conclusão da análise no circuito analógico.</i>	<i>Resultado/Conclusão da análise no circuito digital.</i>
1 – Variar a intensidade de brilho de uma das cores que compõem o RGB	Resultado Desfavorável: Foi realizada simulação de circuito em software que demonstrou que se o circuito fosse implementado conforme planejado haveria diferenças no brilho de cada LED que dificultariam sua implementação. Resultado Favorável: A curva de crescimento do brilho é linear, fazendo com que o LED não apresente cintilação.	Resultado Desfavorável: O LED apresenta cintilação quando ligado em intensidade de brilho entre o valor de alimentação do microcontrolador e o terra do circuito. Porém esta cintilação não pode ser vista a olho nú. Resultado Favorável: O circuito é pequeno e com poucas ligações o que atente em plenitude os objetivos a serem alcançados.
2 – Ao juntar as três cores que compõem a tríade conseguir apresentar a cor branca	Resultado Desfavorável: O circuito apresenta um crescimento de brilho proporcional muito rápido. Para regular esta velocidade seria necessário outro circuito de controle que tornaria o projeto mais oneroso e com níveis de consumo elétrico e emissão	Resultado Favorável: A técnica de PWM é muito eficiente pois com níveis gradativos de intensidade de brilho controlado é possível o controle individual de cada cor de LED sem dispositivos externos ao microcontrolador. Garantindo menor tamanho ao circuito

<i>Objetivo a ser alcançado na etapa de avaliação</i>	<i>Resultado/Conclusão da análise no circuito analógico.</i>	<i>Resultado/Conclusão da análise no circuito digital.</i>
	térmica mais altos em relação ao projeto digital.	total.
3 – Conseguir apresentar as 16 (dezesseis) cores básicas em 1 ponto.	Resultado Desfavorável: Durante esta fase de análise foi constatado documento do fabricante dos LEDs utilizados que recomenda que não seja utilizado o método de variação de tensão. Portanto esta fase não foi realizada e a implementação do circuito por processo analógico foi abortado.	Resultado Favorável: Foi obtido o controle individual de cada cor da tríade permitindo a apresentação das 16 cores pela primeira tríade.

[Fonte: O autor]

Conforme apresentado, o circuito digital apresentou-se como a melhor opção para o desenvolvimento do primeiro protótipo do painel de LEDs.

6.4 - O Protótipo da Tela de LEDs

Como a análise dos circuitos apontaram para o desenvolvimento do projeto com a maximização do estado sólido, continuamos o desenvolvimento deste projeto com as próximas etapas para a construção da tela. Elencamos abaixo as próximas etapas.

- 1 – Montar o esquema do circuito para apresentar 9 *pixels* (compostos por 3 LEDs, na tríade RGB).
- 2 – Montar protótipo conforme o esquema.
- 3 – Criar código para o microcontrolador capaz de ligar os 9 *pixels* (27 LEDs) simultaneamente e apresentar todas as 16 cores em intervalos de pelo menos 0,5 segundos.
- 4 – Otimizar código para que seja possível, de forma simples, ligar um pixel qualquer com uma cor qualquer (dentro das 16 válidas).

6.4.1 - Primeira Etapa

Utilizando o software Electronics Workbench 9, elaboramos o desenho do que futuramente se tornará o protótipo da tela. O esquema da figura 36 abaixo está utilizando alguns recursos do software que devem ser explicados para o perfeito entendimento do esquemático.

São utilizados os recursos de barramento (BUS) do software para evitar o cruzamento dos fios de ligação entre os componentes. Podemos observar na figura 36 que existem, por exemplo, dois barramentos denominados BUS1. Um deles está ligado aos pinos RE0, RE1 e RE2 do microcontrolador, enquanto o outro está ligado aos LEDs 25, 26 e 27. Devemos interpretar que o LED 27 está ligado à porta RE0 do microcontrolador e assim sucessivamente. Para cada tríade foi criado um barramento no desenho para facilitar sua interpretação.

Os recursos como o oscilador utilizado e a fonte implementada permanecem inalterados em relação à última etapa de análise do circuito digital.

A figura 36 demonstra o esquema para a construção do protótipo para apresentação.

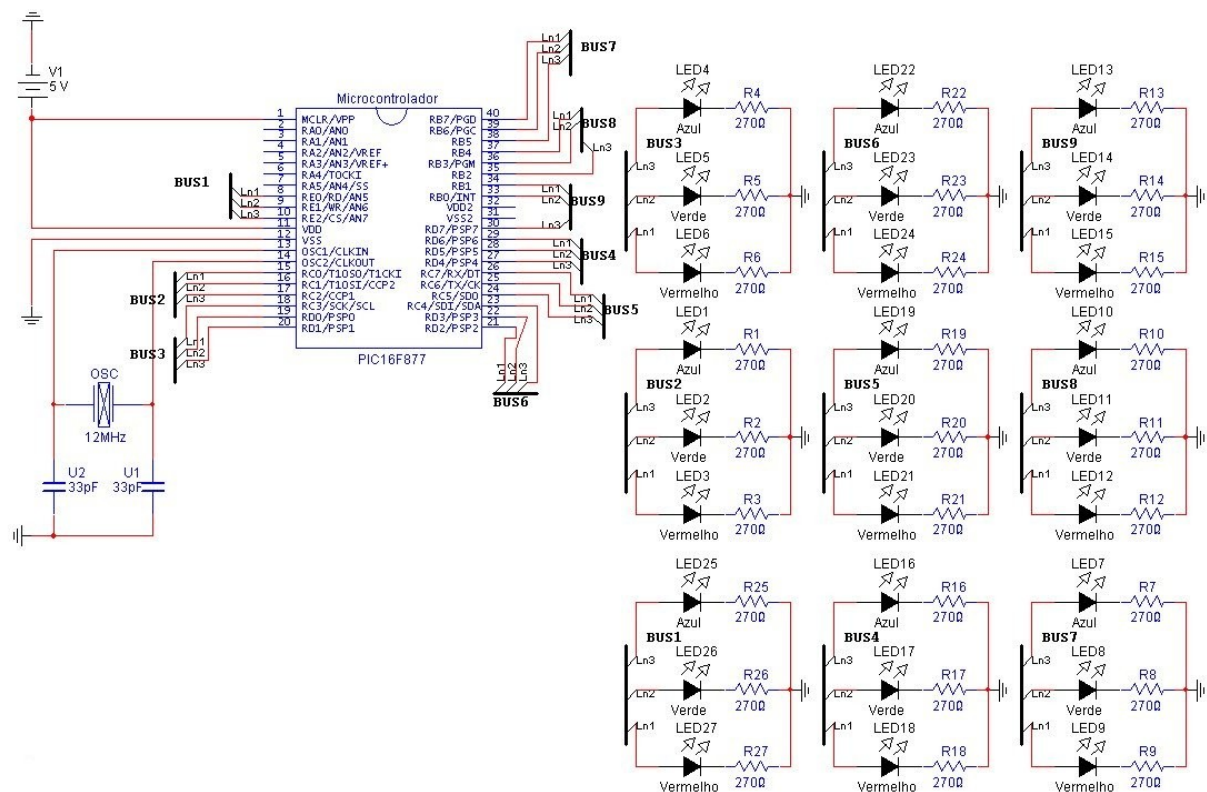


Diagrama esquemático para construção da tela com 3x3 pixels coloridos formados por LEDs de alto brilho

Figura 36 – Esquemático do display 3x3 colorido

[Fonte: O autor]

6.4.2 - Segunda Etapa

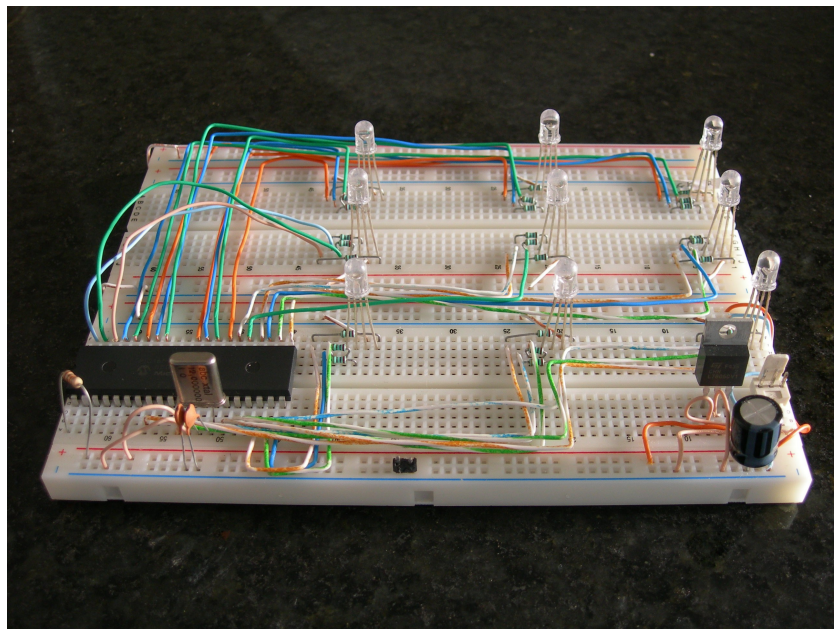
Nesta etapa vamos implementar o *display*, conforme o esquemático elaborado na primeira etapa.

Durante o processo de construção algumas características foram adicionadas na intenção de melhorar o projeto (figura 37).

1 – Desde o início do desenvolvimento deste projeto, a idéia era adquirir no mercado componentes eletrônicos da melhor qualidade possível. A exemplo do que aconteceu durante a análise do circuito digital, que não foi possível a aquisição, no mercado local, de um microcontrolador melhor que o PIC 16F873. Nesta fase de implementação conseguimos adquirir, após muito procurar, substituir os 3 LEDs que compõem a tríade RGB por um LED

que contém 3 núcleos semicondutores e que possuem 4 pinos, sendo um anodo comum aos 3 núcleos e um pino para o catodo de cada cor (conforme datasheet, anexo). Da mesma forma que na análise do circuito digital os núcleos dos LEDs têm intensidade luminosa diferenciada para que seja possível emitir a cor branca (verde e vermelho mais intensos que o azul). Desta forma, estaremos utilizando este componente para cada triade.

2 – Será adicionado a cada LED tricolor um filtro por dois motivos. O mais importante deles é que ao trabalharmos com o protótipo já construído detectamos que, após algumas horas de trabalho, estávamos enxergando LEDs ao fechar os olhos. E o segundo é que, como já afirmado anteriormente, este projeto é para ser visto à distância mínima de 3 (três) metros.



Protótipo do display apresentando 9 *pixels* e desligado. LEDs sem filtro.

Figura 37 – Protótipo Final.

[Fonte: O autor]

Em relação ao esquemático apresentado, algumas alterações na ordem de ligação dos pinos foram feitas no intuito de melhorar a disposição dos fios. Na tabela 6 apresentamos correspondência final dos pinos de alimentação dos LEDs com o microcontrolador.

Tabela 6 – Correspondência de pinos com LEDs.

<i>Pixel / Cor do LED</i>	<i>Porta / Pino</i>
Pixel 1azul	D6
Pixel 1verde	D5
Pixel 1vermelho	D4
Pixel 2azul	B1
Pixel 2verde	B0
Pixel 2vermelho	D7
Pixel 3azul	B4
Pixel 3verde	B3
Pixel 3vermelho	B2
Pixel 4azul	B7
Pixel 4verde	B6
Pixel 4vermelho	B5
Pixel 5azul	C7
Pixel 5verde	C6
Pixel 5vermelho	C5
Pixel 6azul	C4
Pixel 6verde	D3
Pixel 6vermelho	D2
Pixel 7azul	D1
Pixel 7verde	D0
Pixel 7vermelho	C3
Pixel 8azul	E2
Pixel 8verde	E1
Pixel 8vermelho	E0
Pixel 9azul	C2
Pixel 9verde	C1
Pixel 9vermelho	C0

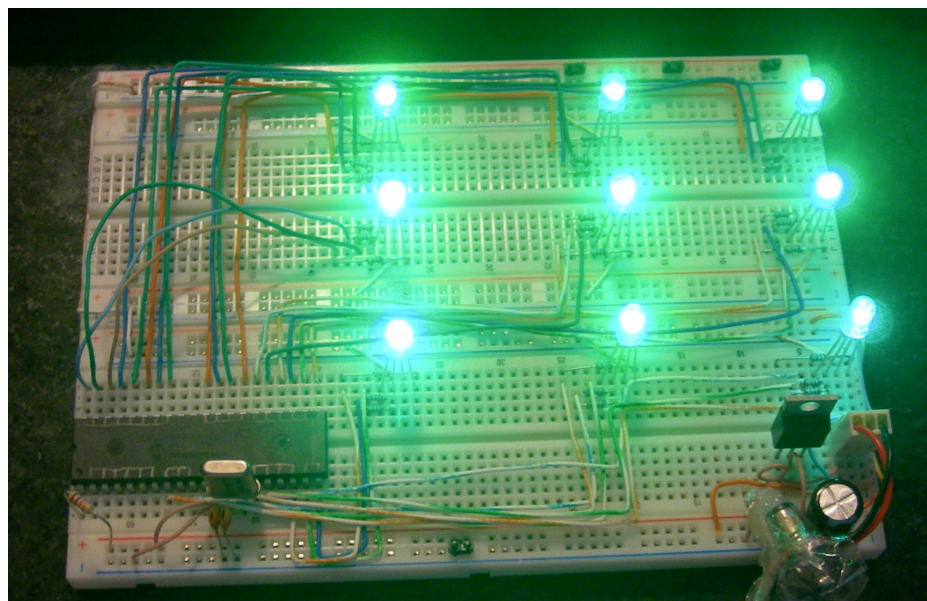
[Fonte: O autor]

6.4.3 - Terceira Etapa

Nesta etapa, trataremos do código, para ligar todos os *pixels* do *display* simultâneamente apresentando as 16 cores e alternando entre elas em pequenos intervalos. Um dos objetivos desta etapa é garantir que o microcontrolador terá memória RAM e FLASH suficientes para executar este programa e fazer constatações com relação a aquecimento e alimentação do circuito. A grosso modo, se o microcontrolador suportar a execução deste código estará apto a rodar diversos códigos para controle das cores e dos *pixels*.

Constatamos que ao compilarmos o código fonte completo com todas as 16 cores o compilador havia informado que a RAM disponível neste microcontrolador não seria suficiente para executar este código fonte. Fizemos a redução do código fonte para 5 cores e o mesmo foi compilado com êxito. Já havíamos planejado uma redução no código fonte para a quarta etapa de desenvolvimento do protótipo, desta forma decidimos por continuar o desenvolvimento.

A figura 38 demonstra a execução do código apresentado no apêndice 4 e comprova o funcionamento do mesmo.



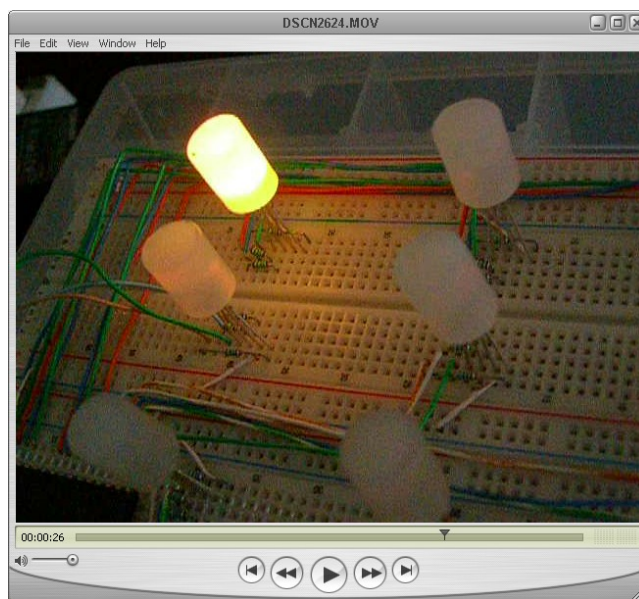
Protótipo do display apresentando 9 *pixels* ligados com a cor verde. LEDs sem filtro

Figura 38 – Protótipo Final Ligado.

[Fonte: O autor]

6.4.4 - Quarta Etapa

A otimização do código constitui em aumentar o controle sobre os *pixels* e suas cores. Conforme código apenso, através da função “TrataPixel(Branco,Pixel_1)” podemos acionar o *pixel* 1 com a cor branca e existe uma função para cada cor, bastando apenas alterar a cor e o pixel para acioná-lo. A figura 39 demonstra a execução do código (em apêndice 5).



Protótipo do *display* apresentando 1 pixel ligado com a cor marrom. Pixel recoberto com filtro para mistura das cores.

Figura 39 – Tríade emitindo cor marrom.

[Fonte: O autor]

6.4.5 - Conclusões da montagem do Protótipo.

Como já esperado, o protótipo atendeu a todas as expectativas e funcionou perfeitamente. Foi constatado que o microcontrolador, mesmo ligado por algumas horas, executando o código da terceira etapa, que é um código robusto, permanece com temperatura pouco acima da temperatura ambiente. Não há cintilação visível entre as cores da tríade RGB e também não há cintilação entre os pixel, quando todos acionados. Isto demonstra que o microcontrolador tem velocidade suficiente para executar os códigos mais desotimizados e manter o controle da intensidade do brilho de cada cor da tríade, em perfeita condição de ser apresentado para quem observa seus pontos.

A maior dificuldade encontrada nestas fases foram a implementação da função que controla qual pino do microcontrolador será posto em nível baixo ou alto sem alterar os outros pinos da mesma porta. O compilador CCS possui as funções `output_high()` e `output_low()` que

se destinam a este fim, porém estas funções não aceitam a passagem de parâmetros em variáveis. A solução foi criar uma função equivalente através do uso de ponteiros, movimentação de bits e operações aritméticas com bits.

CAPÍTULO 7 – CONCLUSÃO

7.1 – Conclusões

Com relação à análise entre o circuito analógico e o digital constatamos que para este projeto seria melhor prosseguir com o desenvolvimento do circuito de forma digital. Para chegar a esta conclusão avaliamos diversas características, como o tamanho físico do circuito final, a dissipação térmica do circuito nas duas implementações e outras. No entanto, não descartamos que em outras condições de desenvolvimento, o circuito analógico não seja melhor aplicado.

A proposta inicial foi atingida. Conseguimos apresentar um protótipo de tela de 3x3 pixels (27 LEDs) que consegue emitir inicialmente 16 cores para cada pixel que podem ser controlados por uma simples função de código. A técnica do PWM atendeu acima do esperado, não tivemos problema algum com o seu uso. Inclusive a mesma é aconselhada pelos próprios fabricantes de LEDs para obter a variação do brilho.

Com relação a questão do consumo elétrico não foram realizados testes neste sentido, porém se compararmos os tipos e quantidades de componentes utilizados entre a implementação do circuito analógico e digital, bem como sua dissipação térmica individual, conforme datasheets dos capacitores, resistores e amplificadores operacionais, podemos afirmar que o circuito digital apresentaria uma menor relação de troca de calor com o ambiente o que normalmente significa um consumo elétrico mais otimizado.

7.2 – Dificuldades encontradas

A maior dificuldade deste projeto, acreditamos que tenha sido encontrar componentes eletrônicos para aquisição. Por duas etapas fundamentais tivemos que comprar componentes com menos recursos do que tínhamos idealizado para não interromper o desenvolvimento do

projeto. Uma destas etapas foi a troca do microcontrolador PIC 16F873 pelo PIC 16F877, onde desde o início já havíamos planejado a utilização do PIC 16F877. A outra etapa de dificuldade foi a obtenção do LED de 3 cores, que apesar de termos planejado a utilização de 3 LEDs individuais, já sabíamos de sua existência e só conseguimos adquiri-lo no último mês deste projeto.

Outra dificuldade que vale a pena citar é a dificuldade de encontrar informações sobre LEDs. Como o LED é um produto, quase tudo que se encontra, na verdade, está em livros de semicondutores. Características como brilho, curvas de luminosidade e outros, temos que confiar em informações de fabricantes. Alguns dados como a quantidade de LEDs com defeito de fabricação simplesmente não foram obtidos.

7.3 – Trabalhos futuros

Diversas linhas de pesquisa podem ser criadas a partir do trabalho aqui apresentado. Dentre elas podemos destacar:

- Implementar a comunicação serial entre diversos microcontroladores de forma que todos possam receber informações simultaneamente e apresentá-las.
- Aumentar a quantidade de *pixels* do sistema, permitindo que uma tela maior seja implementada.
- Melhorar o controle das cores com um ajuste mais fino. Aumentar a amostragem entre os níveis de PWM.
- Aumentar a quantidade de cores do sistema, para pelo menos 256 cores.
- Integrar pelo menos 4 microcontroladores para formar uma tela de 6x6 pixels (108 LEDs).
- Melhorar a codificação, para que fique mais eficiente permitindo um melhor aproveitamento da velocidade do microcontrolador.
- Realizar controle individual da curva de brilho de cada LED por software ou pela adição de circuito adicional ao painel.

Referências Bibliográficas

BARR, Michael. *"Pulse Width Modulation" Embedded Systems Programming. Elkrige, MD: Netrino, 2001. p. 103-104. Disponível em: <<http://www.netrino.com/Publications/Glossary/PWM.php>> e <http://pt.wikipedia.org/wiki/Modula%C3%A7%C3%A3o_por_largura_de_pulso>. Acesso em: 15 jun 2007.*

BERTULANI, Carlos. *Ensino de Física a Distância, Luz e Cor*. Rio de Janeiro, *sin data*. Disponível em: <<http://www.if.ufrj.br/teaching/luz/cor.html>>. Acesso em: 15 jun 2007.

BOWDEN, Bill. *Bowden's Hobby Circuits*. USA: Bowden, 2006. Disponível em: <http://ourworld.compuserve.com/homepages/Bill_Bowden/page5.htm#eyes.gif>. Acesso em: 15 jun 2007.

CUSINATO, Liur Jorge. *Luminotécnica*. São Leopoldo – RS: UNISINOS, 2006. p. 6

FERREIRA, Aurélio B. H. *Dicionário Aurélio Eletrônico – Século XXI*. Rio de Janeiro – RJ: Editora Nova Fronteira, 1999

FREEDMAN, Roger A.; YOUNG, Hugh D. *Física IV. Tradução: LUIZ, Adir M.* 10.ed. São Paulo: Addison Wesley, 2003. p. 1

GUERRINI, Iria Müller. *Programa Educ@r – Refração: Fundamentos Teóricos*. São Carlos – SP: CDCC USP, 2007. Disponível em: <<http://educar.sc.usp.br/optica/refracao.htm>>. Acesso em: 15 jun 2007.

GURGEL, Maria Fernanda do C. *Método Periódico Para Interpretar Propriedades dos Materiais*. São Carlos – SP: UFSCAR, 2005. Disponível em: <http://www.bdtf.ufscar.br/tde_arquivos/18/TDE-2005-05-23T12:48:47Z-678/Publico/TeseMFCG.pdf>. Acesso em: 13 jun 2007. p. 17

IEEE VIRTUAL MUSEUM, *What is a Semiconductor?*. New Brunswick – NJ, *sin data*. Disponível em: <<http://ieee-virtual-museum.org/collection/tech.php?id=2345928&lid=1>>. Acesso em: 15 jun 2007.

JABBOUR, Ghassan.; LI, Jian.; WILLIAMS, Evan. Et al. *Excimer-Based White Phosphorescent Organic Light-Emitting Diodes with Nearly 100% Internal Quantum Efficiency - Advanced Materials*. - Tempe, AZ: Arizona State University, 2006. p. 197-202. v. 19

KOCH, Kristin. *Current Biology*. Pennsylvania: University of Pennsylvania School of Medicine, 2007. Disponível em: <<http://www.physorg.com/news73156830.html>>. Acesso em: 29 jun 2007.

LAVINIA, Nicolás C.; SOUZA, David J. *Conectando o PIC 16F877A, Recursos Avançados*. 2.ed. São Paulo: Editora Erica, 2005.

LUCAS, Luís Alberto. *Osciladores*. Curitiba – PR: UTFPR – Universidade Tecnológica Federal do Paraná, 2006. Disponível em: <<http://www.daeln.cefetpr.br/~lalucas/>> e <<http://www.daeln.cefetpr.br/~lalucas/Disciplinas/Eln3/Turma%20de%20F%82rias/Trabalhos/Osciladores%20a%20cristal.doc>>. Acesso em: 15 jun 2007.

MALVINO, Albert Paul. *Eletrônica*. 4.ed. 1.v. São Paulo – SP: Pearson Education do Brasil, 1997.

MARQUES, Angelo Eduardo B.; CHOUERI, Salomão Júnior.; CRUZ, Eduardo César A. *Dispositivos Semicondutores: Diodos e Transistores*. 6.ed. São Paulo: Editora Érica, 2001.

MELLO, Hilton A.; INTRATOR, Edmond. *Dispositivos Semicondutores*. 4.ed. Rio de Janeiro – RJ: Editora Livros Técnicos e Científicos, 1980.

MORENO, Ivan.; MOLINAR, Luis M. *Color uniformity of the light distribution from several cluster configurations of multicolor LEDs*. Zacatecas, Mexico: Universidad Autonoma de Zacatecas, Mexico. 2005.

NICHIA CORPORATION. *Controlling Luminous Intensity of LEDs*. Tokyo: Nichia, 2004. Disponível em: <<http://www.nichia.co.jp/specification/appli/controlling.pdf>>. Acesso em: 15 jun 2007.

PEREIRA, Fábio. Microcontroladores PIC: Programação em C. 3.ed. São Paulo: Editora Érica, 2004.

REZENDE, Sergio M. *Materiais e Dispositivos Eletrônicos*. 2.ed. São Paulo: Editora Livraria da Física - Departamento de Física da Universidade Federal de Pernambuco, 2004.

SALLES, Filipe. *Projeto Mnemocine, Memória e Imagem*. São Paulo, *sin data*. Disponível em: <<http://www.mnemocine.com.br/cinema/cinematografia1.htm>>. Acesso em: 13 jun 2007.

Autor do texto: Professor de fotografia na FAAP e mestre em Comunicação e Semiótica na PUC/SP

SANTOS, C.A. *Fundamentos do Modelo Atômico de Bohr*. Porto Alegre – RS: UFRGS, *sin data*. Disponível em: <<http://www.if.ufrgs.br/tex/fis142/modelobohr/modelobohrconc.html>>. Acesso em: 15 jun 2007.

SILVESTRE, Ana Paula C. *Restauração de Imagens Digitais Utilizando Algoritmo de Lucy-Richardson e Deconvolução Cega*. Brasília: Centro Universitário de Brasília – Uniceub, 2005.

SOUZA, David J. *Desbravando o PIC – Ampliado e atualizado para PIC16F628A*. 8.ed. São Paulo: Editora Érica, 2005. p. 67-70.

STRINGFELLOW, G. B.; CRAFT, M. George. *High Brightness Light Emitting Diodes*. *Treatise Editors: WILLARDSON, R. K. and WEBER, Eicke R.* 1.ed. San Diego – CA – USA: Academic Press, 15 jan 1997. 469p. (*Semiconductors and Semimetals*, 48) p. 01 - 03, 180 - 186.

YOUNG, Hugh D.; FREEDMAN, Roger A. *Física II – Termodinâmica e ondas*. Tradução: LUIZ, Adir M. 10.ed. São Paulo: Addison Wesley, 2003.

Apêndice 1

Códigos utilizados em C para realização deste projeto

- Programa para microcontrolador PIC 16F873 utilizado na primeira etapa do circuito digital

```
//
// Varia a intensidade de brilho do LED ligado a porta RB7 em 5 níveis diferentes
//
// Arquivo PWM1LED.c          Alessandro S. Ferreira  21/09/2006
//

#include <16f873.h>
#include delay(clock=12000000)
#include HS,NOWDT,PUT,BROWNOUT,NOLVP

main()
{
    int tempo_h;
    int tempo_l;
    int salto;
    int pula;
    pula=5000;
    while (true)
    {
        // LED permanece 100% do tempo desligado
        tempo_h=0;          //atribui o tempo em que fica nivel alto
        tempo_l=20;         //atribui o tempo em que fica nivel baixo
        salto=0;
        while (salto<pula)
        {
            output_high(PIN_B7);          //seta o pino RB7 nivel alto
            delay_ms(tempo_h);             //aguarda 10ms
            output_low(PIN_B7);            //seta o pino RB7 nivel baixo
            delay_ms(tempo_l);             //aguarda 90ms
            salto++;
        }
        // LED permanece 5% do tempo ligado
        tempo_h=1;          //atribui o tempo em que fica nivel alto
        tempo_l=19;         //atribui o tempo em que fica nivel baixo
        salto=0;
        while (salto<pula)
        {
            output_high(PIN_B7);          //seta o pino RB7 nivel alto
            delay_ms(tempo_h);             //aguarda 10ms
            output_low(PIN_B7);            //seta o pino RB7 nivel baixo
            delay_ms(tempo_l);             //aguarda 90ms
            salto++;
        }
        // LED permanece 50% do tempo ligado
        salto=0;
        tempo_h=10;         //atribui o tempo em que fica nivel alto
        tempo_l=10;         //atribui o tempo em que fica nivel baixo
        while (salto<pula)
        {
            output_high(PIN_B7);          //seta o pino RB7 nivel alto
            delay_ms(tempo_h);             //aguarda 50ms
            output_low(PIN_B7);            //seta o pino RB7 nivel baixo
            delay_ms(tempo_l);             //aguarda 50ms
            salto++;
        }
        // LED permanece 75% do tempo ligado
        salto=0;
        tempo_h=15;         //atribui o tempo em que fica nivel alto
        tempo_l=5;          //atribui o tempo em que fica nivel baixo
        while (salto<pula)
        {
            output_high(PIN_B7);          //seta o pino RB7 nivel alto
            delay_ms(tempo_h);             //aguarda 75ms
            output_low(PIN_B7);            //seta o pino RB7 nivel baixo
```



```

        delay_ms(tempo_l);          //aguarda 25ms
        salto++;
    }
    // LED permanece 100% do tempo ligado
    salto=0;
    tempo_h=20;          //atribui o tempo em que fica nivel alto
    tempo_l=0;           //atribui o tempo em que fica nivel baixo
    while (salto<pula)
    {
        output_high(PIN_B7);        //seta o pino RB7 nivel alto
        delay_ms(tempo_h);          //aguarda 75ms
        output_low(PIN_B7);         //seta o pino RB7 nivel baixo
        delay_ms(tempo_l);          //aguarda 25ms
        salto++;
    }
}
// ----- FIM -----

```

Apêndice 2

- Programa para microcontrolador PIC 16F873 utilizado na segunda etapa do circuito digital

```
//
// Varia a intensidade de brilho de 3LEDs ligados as portas RB7,RB6 e RB5 em 5 níveis
// diferentes de igual intensidade para cada LED.
//
// Arquivo PWM3LED.c          Alessandro S. Ferreira  24/03/2007
//

#include <16f873.h>
#include delay(clock=12000000)
#include HS,NOWDT,PUT,BROWNOUT,NOLVP

main()
{
    int tempo_h;
    int tempo_l;
    int salto;
    int pula;
    pula=5000;
    while (true)
    {
        // LEDs permanecem 100% do tempo desligados
        tempo_h=0;          //atribui o tempo em que fica nivel alto
        tempo_l=20;         //atribui o tempo em que fica nivel baixo
        salto=0;
        while (salto<pula)
        {
            output_high(PIN_B7);      //seta o pino RB7 nivel alto
            output_high(PIN_B6);      //seta o pino RB6 nivel alto
            output_high(PIN_B5);      //seta o pino RB6 nivel alto
            delay_ms(tempo_h);        //aguarda 0ms
            output_low(PIN_B7);       //seta o pino RB7 nivel baixo
            output_low(PIN_B6);       //seta o pino RB6 nivel baixo
            output_low(PIN_B5);       //seta o pino RB5 nivel baixo
            delay_ms(tempo_l);        //aguarda 20ms
            salto++;
        }
        // LEDs permanecem 5% do tempo ligados
        tempo_h=1;          //atribui o tempo em que fica nivel alto
        tempo_l=19;         //atribui o tempo em que fica nivel baixo
        salto=0;
        while (salto<pula)
        {
            output_high(PIN_B7);      //seta o pino RB7 nivel alto
            output_high(PIN_B6);      //seta o pino RB6 nivel alto
            output_high(PIN_B5);      //seta o pino RB6 nivel alto
            delay_ms(tempo_h);        //aguarda 1ms
            output_low(PIN_B7);       //seta o pino RB7 nivel baixo
            output_low(PIN_B6);       //seta o pino RB6 nivel baixo
            output_low(PIN_B5);       //seta o pino RB5 nivel baixo
            delay_ms(tempo_l);        //aguarda 19ms
            salto++;
        }
        // LEDs permanece 50% do tempo ligados
        salto=0;
        tempo_h=10;         //atribui o tempo em que fica nivel alto
        tempo_l=10;         //atribui o tempo em que fica nivel baixo
        while (salto<pula)
        {
            output_high(PIN_B7);      //seta o pino RB7 nivel alto
            output_high(PIN_B6);      //seta o pino RB6 nivel alto
            output_high(PIN_B5);      //seta o pino RB6 nivel alto
            delay_ms(tempo_h);        //aguarda 10ms
            output_low(PIN_B7);       //seta o pino RB7 nivel baixo
            output_low(PIN_B6);       //seta o pino RB6 nivel baixo
            output_low(PIN_B5);       //seta o pino RB5 nivel baixo
            delay_ms(tempo_l);        //aguarda 10ms
            salto++;
        }
    }
}
```

```

// LEDs permanece 75% do tempo ligados
salto=0;
tempo_h=15;          //atribui o tempo em que fica nivel alto
tempo_l=5;           //atribui o tempo em que fica nivel baixo
while (salto<pula)
{
    output_high(PIN_B7);      //seta o pino RB7 nivel alto
    output_high(PIN_B6);      //seta o pino RB6 nivel alto
    output_high(PIN_B5);      //seta o pino RB6 nivel alto
    delay_ms(tempo_h);        //aguarda 15ms
    output_low(PIN_B7);       //seta o pino RB7 nivel baixo
    output_low(PIN_B6);       //seta o pino RB6 nivel baixo
    output_low(PIN_B5);       //seta o pino RB5 nivel baixo
    delay_ms(tempo_l);        //aguarda 5ms
    salto++;
}
// LEDs permanece 100% do tempo ligados
salto=0;
tempo_h=20;          //atribui o tempo em que fica nivel alto
tempo_l=0;           //atribui o tempo em que fica nivel baixo
while (salto<pula)
{
    output_high(PIN_B7);      //seta o pino RB7 nivel alto
    output_high(PIN_B6);      //seta o pino RB6 nivel alto
    output_high(PIN_B5);      //seta o pino RB6 nivel alto
    delay_ms(tempo_h);        //aguarda 20ms
    output_low(PIN_B7);       //seta o pino RB7 nivel baixo
    output_low(PIN_B6);       //seta o pino RB6 nivel baixo
    output_low(PIN_B5);       //seta o pino RB5 nivel baixo
    delay_ms(tempo_l);        //aguarda 0ms
    salto++;
}
}
}
// ----- FIM -----

```

Apêndice 3

- Programa para microcontrolador PIC 16F877 utilizado na terceira etapa do circuito digital

```
//
// Aplica padroes de cores CGA na ordem em 1 pixel
//
// Arquivo CGA16cor1pix.c          Alessandro S. Ferreira   26/05/2007
//

#include <16f877.h>
#define delay(clock=12000000)
#define HS,NOWDT,PUT,BROWNOUT,NOLVP

#define pix01b      pin_b7
#define pix01g      pin_b6
#define pix01r      pin_b5

main()
{
    int tempo_borda;
    int salto;
    int pula;
    pula=2000;
    while (true)
    {
        tempo_borda=1; //atribui o tempo em que varia de nivel alto para baixo e vice versa

        salto=0;
        while (salto<pula)
        {
            // Nivel de cor 00(R),00(G),00(B) -> Preto (0)
            output_low(pix01r); //seta o pixel 01 vermelho nivel baixo
            output_low(pix01g); //seta o pixel 01 verde nivel baixo
            output_low(pix01b); //seta o pixel 01 azul nivel baixo
            delay_ms(tempo_borda); //aguarda tempo em ms

            output_low(pix01r); //seta o pixel 01 vermelho nivel baixo
            output_low(pix01g); //seta o pixel 01 verde nivel baixo
            output_low(pix01b); //seta o pixel 01 azul nivel baixo
            delay_ms(tempo_borda); //aguarda tempo em ms

            output_low(pix01r); //seta o pixel 01 vermelho nivel baixo
            output_low(pix01g); //seta o pixel 01 verde nivel baixo
            output_low(pix01b); //seta o pixel 01 azul nivel baixo
            delay_ms(tempo_borda); //aguarda tempo em ms

            salto++;
        }
        salto=0;
        while (salto<pula)
        {
            // Nivel de cor 00(R),00(G),AA(B) -> Azul (1)
            output_low(pix01r); //seta o pixel 01 vermelho nivel baixo
            output_low(pix01g); //seta o pixel 01 verde nivel baixo
            output_high(pix01b); //seta o pixel 01 azul nivel alto
            delay_ms(tempo_borda); //aguarda tempo em ms

            output_low(pix01r); //seta o pixel 01 vermelho nivel baixo
            output_low(pix01g); //seta o pixel 01 verde nivel baixo
            output_high(pix01b); //seta o pixel 01 azul nivel alto
            delay_ms(tempo_borda); //aguarda tempo em ms

            output_low(pix01r); //seta o pixel 01 vermelho nivel baixo
            output_low(pix01g); //seta o pixel 01 verde nivel baixo
            output_low(pix01b); //seta o pixel 01 azul nivel baixo
            delay_ms(tempo_borda); //aguarda tempo em ms

            salto++;
        }
    }
}
```

```

}
salto=0;
while (salto<pula)
{
    // Nivel de cor 00(R),AA(G),00(B) -> Verde (2)
    output_low(pix01r);    //seta o pixel 01 vermelho nivel baixo
    output_high(pix01g);    //seta o pixel 01 verde nivel alto
    output_low(pix01b);    //seta o pixel 01 azul nivel baixo
    delay_ms(tempo_borda);    //aguarda tempo em ms

    output_low(pix01r);    //seta o pixel 01 vermelho nivel baixo
    output_high(pix01g);    //seta o pixel 01 verde nivel alto
    output_low(pix01b);    //seta o pixel 01 azul nivel baixo
    delay_ms(tempo_borda);    //aguarda tempo em ms

    output_low(pix01r);    //seta o pixel 01 vermelho nivel baixo
    output_low(pix01g);    //seta o pixel 01 verde nivel baixo
    output_low(pix01b);    //seta o pixel 01 azul nivel baixo
    delay_ms(tempo_borda);    //aguarda tempo em ms

    salto++;
}
salto=0;
while (salto<pula)
{
    // Nivel de cor 00(R),AA(G),AA(B) -> Cyan (3)
    output_low(pix01r);    //seta o pixel 01 vermelho nivel baixo
    output_high(pix01g);    //seta o pixel 01 verde nivel alto
    output_high(pix01b);    //seta o pixel 01 azul nivel alto
    delay_ms(tempo_borda);    //aguarda tempo em ms

    output_low(pix01r);    //seta o pixel 01 vermelho nivel baixo
    output_high(pix01g);    //seta o pixel 01 verde nivel alto
    output_high(pix01b);    //seta o pixel 01 azul nivel alto
    delay_ms(tempo_borda);    //aguarda tempo em ms

    output_low(pix01r);    //seta o pixel 01 vermelho nivel baixo
    output_low(pix01g);    //seta o pixel 01 verde nivel baixo
    output_low(pix01b);    //seta o pixel 01 azul nivel baixo
    delay_ms(tempo_borda);    //aguarda tempo em ms

    salto++;
}
salto=0;
while (salto<pula)
{
    // Nivel de cor AA(R),00(G),00(B) -> Vermelho (4)
    output_high(pix01r);    //seta o pixel 01 vermelho nivel alto
    output_low(pix01g);    //seta o pixel 01 verde nivel baixo
    output_low(pix01b);    //seta o pixel 01 azul nivel baixo
    delay_ms(tempo_borda);    //aguarda tempo em ms

    output_high(pix01r);    //seta o pixel 01 vermelho nivel alto
    output_low(pix01g);    //seta o pixel 01 verde nivel baixo
    output_low(pix01b);    //seta o pixel 01 azul nivel baixo
    delay_ms(tempo_borda);    //aguarda tempo em ms

    output_low(pix01r);    //seta o pixel 01 vermelho nivel baixo
    output_low(pix01g);    //seta o pixel 01 verde nivel baixo
    output_low(pix01b);    //seta o pixel 01 azul nivel baixo
    delay_ms(tempo_borda);    //aguarda tempo em ms

    salto++;
}
salto=0;
while (salto<pula)
{
    // Nivel de cor AA(R),00(G),AA(B) -> Magenta (5)
    output_high(pix01r);    //seta o pixel 01 vermelho nivel alto
    output_low(pix01g);    //seta o pixel 01 verde nivel baixo
    output_high(pix01b);    //seta o pixel 01 azul nivel alto
    delay_ms(tempo_borda);    //aguarda tempo em ms

    output_high(pix01r);    //seta o pixel 01 vermelho nivel alto
    output_low(pix01g);    //seta o pixel 01 verde nivel baixo
    output_high(pix01b);    //seta o pixel 01 azul nivel alto
    delay_ms(tempo_borda);    //aguarda tempo em ms

    output_low(pix01r);    //seta o pixel 01 vermelho nivel baixo

```

```

        output_low(pix01g);          //seta o pixel 01 verde nivel baixo
        output_low(pix01b);          //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);        //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor AA(R),55(G),00(B) -> Marrom (6)
        output_high(pix01r);          //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);          //seta o pixel 01 verde nivel alto
        output_low(pix01b);           //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);        //aguarda tempo em ms

        output_high(pix01r);          //seta o pixel 01 vermelho nivel alto
        output_low(pix01g);           //seta o pixel 01 verde nivel baixo
        output_low(pix01b);           //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);        //aguarda tempo em ms

        output_low(pix01r);           //seta o pixel 01 vermelho nivel baixo
        output_low(pix01g);           //seta o pixel 01 verde nivel baixo
        output_low(pix01b);           //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);        //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor AA(R),AA(G),AA(B) -> Cinza (7)
        output_high(pix01r);          //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);          //seta o pixel 01 verde nivel alto
        output_high(pix01b);          //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);        //aguarda tempo em ms

        output_high(pix01r);          //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);          //seta o pixel 01 verde nivel alto
        output_high(pix01b);          //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);        //aguarda tempo em ms

        output_low(pix01r);           //seta o pixel 01 vermelho nivel baixo
        output_low(pix01g);           //seta o pixel 01 verde nivel baixo
        output_low(pix01b);           //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);        //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor 55(R),55(G),55(B) -> Cinza Escuro (8)
        output_high(pix01r);          //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);          //seta o pixel 01 verde nivel alto
        output_high(pix01b);          //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);        //aguarda tempo em ms

        output_low(pix01r);           //seta o pixel 01 vermelho nivel baixo
        output_low(pix01g);           //seta o pixel 01 verde nivel baixo
        output_low(pix01b);           //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);        //aguarda tempo em ms

        output_low(pix01r);           //seta o pixel 01 vermelho nivel baixo
        output_low(pix01g);           //seta o pixel 01 verde nivel baixo
        output_low(pix01b);           //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);        //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor 55(R),55(G),FF(B) -> Azul Claro (9)
        output_high(pix01r);          //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);          //seta o pixel 01 verde nivel alto
        output_high(pix01b);          //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);        //aguarda tempo em ms

        output_low(pix01r);           //seta o pixel 01 vermelho nivel baixo

```

```

        output_low(pix01g);          //seta o pixel 01 verde nivel baixo
        output_high(pix01b);         //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);       //aguarda tempo em ms

        output_low(pix01r);          //seta o pixel 01 vermelho nivel baixo
        output_low(pix01g);          //seta o pixel 01 verde nivel baixo
        output_high(pix01b);         //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);       //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor 55(R),FF(G),55(B) -> Verde Claro (10)
        output_high(pix01r);         //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);         //seta o pixel 01 verde nivel alto
        output_high(pix01b);         //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);       //aguarda tempo em ms

        output_low(pix01r);          //seta o pixel 01 vermelho nivel baixo
        output_high(pix01g);         //seta o pixel 01 verde nivel alto
        output_low(pix01b);          //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);       //aguarda tempo em ms

        output_low(pix01r);          //seta o pixel 01 vermelho nivel baixo
        output_high(pix01g);         //seta o pixel 01 verde nivel alto
        output_low(pix01b);          //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);       //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor 55(R),FF(G),FF(B) -> Cyan Claro (11)
        output_high(pix01r);         //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);         //seta o pixel 01 verde nivel alto
        output_high(pix01b);         //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);       //aguarda tempo em ms

        output_low(pix01r);          //seta o pixel 01 vermelho nivel baixo
        output_high(pix01g);         //seta o pixel 01 verde nivel alto
        output_high(pix01b);         //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);       //aguarda tempo em ms

        output_low(pix01r);          //seta o pixel 01 vermelho nivel baixo
        output_high(pix01g);         //seta o pixel 01 verde nivel alto
        output_high(pix01b);         //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);       //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor FF(R),55(G),55(B) -> Vermelho Claro (12)
        output_high(pix01r);         //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);         //seta o pixel 01 verde nivel alto
        output_high(pix01b);         //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);       //aguarda tempo em ms

        output_high(pix01r);         //seta o pixel 01 vermelho nivel alto
        output_low(pix01g);          //seta o pixel 01 verde nivel baixo
        output_low(pix01b);          //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);       //aguarda tempo em ms

        output_high(pix01r);         //seta o pixel 01 vermelho nivel alto
        output_low(pix01g);          //seta o pixel 01 verde nivel baixo
        output_low(pix01b);          //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);       //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor FF(R),55(G),FF(B) -> Magenta Claro(13)
        output_high(pix01r);         //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);         //seta o pixel 01 verde nivel alto

```

```

        output_high(pix01b);        //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);      //aguarda tempo em ms

        output_high(pix01r);        //seta o pixel 01 vermelho nivel alto
        output_low(pix01g);         //seta o pixel 01 verde nivel baixo
        output_high(pix01b);        //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);      //aguarda tempo em ms

        output_high(pix01r);        //seta o pixel 01 vermelho nivel alto
        output_low(pix01g);         //seta o pixel 01 verde nivel baixo
        output_high(pix01b);        //seta o pixel 01 azul nivel high
        delay_ms(tempo_borda);      //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor FF(R),FF(G),55(B) -> Amarelo (14)
        output_high(pix01r);        //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);        //seta o pixel 01 verde nivel alto
        output_high(pix01b);        //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);      //aguarda tempo em ms

        output_high(pix01r);        //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);        //seta o pixel 01 verde nivel alto
        output_low(pix01b);         //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);      //aguarda tempo em ms

        output_high(pix01r);        //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);        //seta o pixel 01 verde nivel alto
        output_low(pix01b);         //seta o pixel 01 azul nivel baixo
        delay_ms(tempo_borda);      //aguarda tempo em ms

        salto++;
    }
    salto=0;
    while (salto<pula)
    {
        // Nivel de cor FF(R),FF(G),FF(B) -> Branco (15)
        output_high(pix01r);        //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);        //seta o pixel 01 verde nivel alto
        output_high(pix01b);        //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);      //aguarda tempo em ms

        output_high(pix01r);        //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);        //seta o pixel 01 verde nivel alto
        output_high(pix01b);        //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);      //aguarda tempo em ms

        output_high(pix01r);        //seta o pixel 01 vermelho nivel alto
        output_high(pix01g);        //seta o pixel 01 verde nivel alto
        output_high(pix01b);        //seta o pixel 01 azul nivel alto
        delay_ms(tempo_borda);      //aguarda tempo em ms

        salto++;
    }
}
}
// ----- FIM -----

```


Apêndice 4

- Programa para microcontrolador PIC 16F877 utilizado na terceira etapa do protótipo do circuito final.

```
//
// Aplica padrao de cor CGA para todos os PIXELS
//
// Arquivo PWMAllLEDs.c      Alessandro S. Ferreira  20/04/2007
//

#include <16f877.h>
#include delay(clock=12000000)
#include XT,NOWDT,PUT,BROWNOUT,NOLVP

#define pix01b      pin_d6
#define pix01g      pin_d5
#define pix01r      pin_d4

#define pix02b      pin_b1
#define pix02g      pin_b0
#define pix02r      pin_d7

#define pix03b      pin_b4
#define pix03g      pin_b3
#define pix03r      pin_b2

#define pix04b      pin_b7
#define pix04g      pin_b6
#define pix04r      pin_b5

#define pix05b      pin_c7
#define pix05g      pin_c6
#define pix05r      pin_c5

#define pix06b      pin_c4
#define pix06g      pin_d3
#define pix06r      pin_d2

#define pix07b      pin_d1
#define pix07g      pin_d0
#define pix07r      pin_c3

#define pix08b      pin_e2
#define pix08g      pin_e1
#define pix08r      pin_e0

#define pix09b      pin_c2
#define pix09g      pin_c1
#define pix09r      pin_c0

main()
{
    int tempo_borda;
    int salto;
    int pula;
    pula=5000;
    while (true)
    {
        tempo_borda=1;    //atribui o tempo em que varia de nivel alto para baixo e vice versa

        salto=0;
        while (salto<pula)
        {
            // Nivel de cor 00(R),00(G),AA(B) -> Azul (1)
            output_high(pix01b);    //seta o pino corresp. para nivel alto
            output_low(pix01g);     //seta o pino corresp. para nivel baixo
            output_low(pix01r);     //seta o pino corresp. para nivel baixo
            output_high(pix02b);    //seta o pino corresp. para nivel alto
            output_low(pix02g);     //seta o pino corresp. para nivel baixo
            output_low(pix02r);     //seta o pino corresp. para nivel baixo
            output_high(pix03b);    //seta o pino corresp. para nivel alto
```


Apêndice 5

- Programa para microcontrolador PIC 16F877 utilizado na quarta etapa do protótipo do circuito final.

```
//
// Este arquivo contem as funcoes de definem o pixel passado no paramentro com a cor da funcao
//
// Arquivo PWMFuncLEds.c          Alessandro S. Ferreira  10/05/2007
//

#include <16f877.h>
#define delay(clock=12000000)
#define fuses HS,NOWDT,PUT,BROWNOUT,NOLVP
#define fixed_io(b_outputs = PIN_B0, PIN_B1, PIN_B2, PIN_B3, PIN_B4, PIN_B5, PIN_B6, PIN_B7)
#define fixed_io(c_outputs = PIN_C0, PIN_C1, PIN_C2, PIN_C3, PIN_C4, PIN_C5, PIN_C6, PIN_C7)
#define fixed_io(d_outputs = PIN_D0, PIN_D1, PIN_D2, PIN_D3, PIN_D4, PIN_D5, PIN_D6, PIN_D7)
#define fixed_io(e_outputs = PIN_E0, PIN_E1, PIN_E2)
#define byte portb = 6
#define byte portc = 7
#define byte portd = 8
#define byte porte = 9

long pula = 2;
int i,j;
long salto;

int pixel_1[3] = {PIN_D4, PIN_D5, PIN_D6};
int pixel_2[3] = {PIN_B2, PIN_B0, PIN_B1};
int pixel_3[3] = {PIN_D7, PIN_B3, PIN_B4};
int pixel_4[3] = {PIN_B5, PIN_B6, PIN_B7};
int pixel_5[3] = {PIN_C5, PIN_C6, PIN_C7};
int pixel_6[3] = {PIN_D2, PIN_D3, PIN_C4};
int pixel_7[3] = {PIN_C3, PIN_D0, PIN_D1};
int pixel_8[3] = {PIN_E0, PIN_E1, PIN_E2};
int pixel_9[3] = {PIN_C0, PIN_C1, PIN_C2};

// ORDEM: RGB
// 0 = BBB; 1 = BBA; 2 = BAB; 3 = ABB; 4 = BAA; 5 = ABA; 6 = AAB; 7 = AAA ,onde A é Alta e B é
// Baixa

int Preto[3]           = {0,0,0};
int Azul[3]            = {1,1,0};
int Verde[3]           = {2,2,0};
int Cyan[3]            = {4,4,0};
int Vermelho[3]        = {3,3,0};
int Magenta[3]         = {5,5,0};
int Marrom[3]          = {6,3,0};
int Cinza[3]           = {7,7,0};
int CinzaEscuro[3]     = {7,0,0};
int AzulClaro[3]       = {7,1,1};
int VerdeClaro[3]      = {7,2,2};
int CyanClaro[3]       = {7,4,4};
int VermelhoClaro[3]   = {7,3,3};
int MagentaClaro[3]    = {7,5,5};
int Amarelo[3]         = {7,6,6};
int Branco[3]          = {7,7,7};

void BaixaAlta(ba, pino){
    if (ba == 1){
        *(pino/8|0x80) &= ~(1<<(pino&7));
        *(pino/8) |= (1<<(pino&7));
    }else{
        *(pino/8|0x80) &= ~(1<<(pino&7));
        *(pino/8) &= ~(1<<(pino&7));
    }
}

void Limpa()
{
    portb = 0;
    portc = 0;
    portd = 0;
    porte = 0;
}
```



```

}

void TrataPixel(int Cor[3],int Pixel[3]){
    salto = 0;
    while (salto<pula){
        for (i=0;i<3;i++){
            for(j=0;j<3;j++){
                if(Cor[j] == 0)
                {
                    BaixaAlta(0,Pixel[0]); BaixaAlta(0,Pixel[1]); BaixaAlta(0,Pixel[2]);
                }
                if(Cor[j] == 1)
                {
                    BaixaAlta(0,Pixel[0]); BaixaAlta(0,Pixel[1]); BaixaAlta(1,Pixel[2]);
                }
                if(Cor[j] == 2)
                {
                    BaixaAlta(0,Pixel[0]); BaixaAlta(1,Pixel[1]); BaixaAlta(0,Pixel[2]);
                }
                if(Cor[j] == 3)
                {
                    BaixaAlta(1,Pixel[0]); BaixaAlta(0,Pixel[1]); BaixaAlta(0,Pixel[2]);
                }
                if(Cor[j] == 4)
                {
                    BaixaAlta(0,Pixel[0]); BaixaAlta(1,Pixel[1]); BaixaAlta(1,Pixel[2]);
                }
                if(Cor[j] == 5)
                {
                    BaixaAlta(1,Pixel[0]); BaixaAlta(0,Pixel[1]); BaixaAlta(1,Pixel[2]);
                }
                if(Cor[j] == 6)
                {
                    BaixaAlta(1,Pixel[0]); BaixaAlta(1,Pixel[1]); BaixaAlta(0,Pixel[2]);
                }
                if(Cor[j] == 7)
                {
                    BaixaAlta(1,Pixel[0]); BaixaAlta(1,Pixel[1]); BaixaAlta(1,Pixel[2]);
                }
            }
            delay_ms(0);
        }
        salto++;
    }
}

void main()
{
    while (true)
    {
        Limpa();
        TrataPixel(Verde,      Pixel_1);
        Limpa();
        TrataPixel(Amarelo,    Pixel_2);
        Limpa();
        TrataPixel(Marrom,     Pixel_3);
        Limpa();
        TrataPixel(Branco,     Pixel_4);
        Limpa();
        TrataPixel(Azul,       Pixel_5);
        Limpa();
        TrataPixel(Verde,      Pixel_6);
        Limpa();
        TrataPixel(Cyan,       Pixel_7);
        Limpa();
        TrataPixel(Vermelho,   Pixel_8);
        Limpa();
        TrataPixel(Magenta,    Pixel_9);
    }
}
// ----- FIM -----

```

Anexo 1

DATASHEETS E OUTRAS INFORMAÇÕES